



ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

Titulación:

INGENIERÍA TÉCNICA EN INFORMÁTICA DE GESTIÓN

Título del proyecto:

Aplicación Silverlight para sistema de gestión de la calefacción
del Colegio de la Ultzama

Jorge Sanz Diaz

Tutor: Jesús Villadangos Alonso

Pamplona, 7 de Septiembre de 2011



1. Índice

1. Índice	2
2. Resumen.....	3
3. Introducción.....	4
3.1. Introducción a la empresa.....	4
3.2. Antecedentes.....	6
3.3. Objetivos.....	8
3.4. Fases del proyecto.....	9
3.5. Planificación.....	10
4. Desarrollo.....	13
4.1. Adaptación a la empresa.....	13
4.2. Estudio de las alternativas.....	14
4.3. Formación.....	21
4.4. Análisis de requisitos.....	33
4.5. Diseño e implementación.....	41
4.5.1 Diseño modelo E-R.....	41
4.5.2 Diseño de Casos de Uso.....	42
4.5.3 Diseño Protocolo Comunicación.....	75
4.5.4 Implementación.....	84
4.6. Pruebas.....	89
5. Conclusiones y líneas futuras.....	91
5.1. Conclusiones personales.....	91
5.2. Conclusiones técnicas.....	93
5.3. Líneas futuras.....	94
6 Bibliografía.....	96
6.1. Libros de texto.....	96
6.2. Direcciones Web.....	97



2. Resumen

En este documento se recoge la memoria del Proyecto de Fin de Carrera para la obtención del título de Ingeniero Técnico en Informática de Gestión en la Universidad Pública de Navarra.

El proyecto ha sido nombrado como “Aplicación Silverlight para sistema de gestión de la calefacción del colegio de la Ultzama” y ha sido realizado por Jorge Sanz Diaz en las instalaciones de la UPNA para la empresa Lator S.L, en el Centro de I+D de Electrónica y Comunicaciones Jerónimo de Ayanz. El proyecto ha tenido una duración de 9 meses a media jornada, la fecha de inicio tuvo lugar en Noviembre de 2010 y finalizando el mes de Agosto de 2011.

Los encargados de supervisar el correcto funcionamiento de la dinámica del proyecto han sido D. Jesús Villadangos Alonso tutor del proyecto y profesor en la Universidad Pública de Navarra, por otro lado está Patxi Tornaría jefe de la empresa Lator S.L.

El proyecto consiste en una aplicación Silverlight que se encargará de gestionar la calefacción. La finalidad de este proyecto es la implantación en un colegio, pero en un futuro se podrá implantar en domicilios particulares. El proyecto se puede dividir en dos etapas, una primera etapa en la que el desarrollo se hacía sobre un modelo ficticio y una segunda etapa en la que empezamos a desarrollar el programa sobre un modelo real. La idea es mostrar todos los parámetros que se manejan en la calefacción de un colegio de forma atractiva y amigable (interfaz usuario-maquina lo mas sencillo posible, ya que la persona que trabaje con este producto no va a tener unos conocimientos en informática muy extensos).

Para llevar a cabo este proyecto, se han seguido todas las fases del ciclo de vida de un producto: especificación de requisitos, análisis, diseño, implementación, pruebas e implantación.



3. Introducción

3.1 Introducción a la empresa

Lator Ingeniería, empresa fundada en 1995, surge como proyecto innovador de la mano de un grupo de ingenieros industriales con la amplia experiencia en el diseño de maquinaria industrial.



En la actualidad, Lator está compuesto por un equipo de profesionales que conjuga experiencia y sólida preparación en nuevas tecnologías y en el diseño asistido por ordenador en 3D. Asimismo, Lator Ingeniería ha puesto su conocimiento en el montaje de naves e instalaciones industriales al servicio del aprovechamiento energético con el diseño integral de proyectos de Biomasa.

La empresa Lator Ingeniería S.L desarrolla su labor en los sectores de:

- Diseño Industrial
- Naves Industriales
- Desarrollo Sostenible
- Desarrollo de producto e I+D
- Fabricación



Desarrollo Sostenible

Viviendas Eficientes:

Objetivo de promover y facilitar la incorporación del ahorro, la eficiencia energética y las energías renovables en las obras, gestión energética y rehabilitación en la vivienda construida en Navarra.

Desarrollo de productos I+D

El objetivo general de esta área es el de desarrollar producto siguiendo los parámetros marcados por nuestros clientes.

Lator Ingeniería atesora un gran conocimiento del mercado productivo, el cual es puesto al servicio de nuestros clientes en aras a la consecución de un producto que cumpla con criterios tales como: uso, ergonomía, estética, funcionalidad, costo... para así dar una adecuada respuesta al mercado actual.

En el campo de I+D, nuestro servicio va dirigido a obtener una solución a la medida de las necesidades de nuestro cliente, empleando para ello los recursos técnicos que sean necesarios.

Nuestra empresa posee la capacidad técnica y humana necesaria para participar en proyectos liderados por otras empresas o instituciones que requieran nuestro aporte.

Nuestra experiencia nos permite centrarnos en la satisfacción de las necesidades de nuestro cliente, minimizando el coste y plazos de ejecución y tratando de maximizar los beneficios de los resultados del proyecto.



3.2 Antecedentes

Este proyecto surgió a necesidad de continuación de otro proyecto que se llevaba realizando años atrás. Dicho proyecto se empezó a desarrollar a partir de tener un control de forma individualizada de cada radiador. El proyecto consta de un pequeño dispositivo que transforma la energía térmica del agua caliente en energía eléctrica mediante una célula que genera 3 V que son almacenados en un condensador. La electricidad acciona un motor que regula el caudal de entrada de agua caliente al radiador y que controla un termostato de aire. Se instala en unos minutos en el tubo de entrada del agua caliente del radiador y sin necesidad de cables.

Los sensores al estar retroalimentados se encontrarán siempre en un estado de reposo, cada cierto tiempo estos sensores despertarán y enviarán los datos pertinentes.

Los datos serán enviados a unos routers colocados estratégicamente a lo largo de todo el colegio por medio de ondas de radio, estos routers a su vez se encargarán de reenviar esta información al comunicador. Este dispositivo es con el que tendremos que interactuar nosotros. La conexión entre el comunicador y el ordenador será mediante el puerto USB en el que utilizaremos el protocolo de comunicación WinUSB.

El objetivo principal de nuestro proyecto es poder procesar esos datos de la forma más sencilla e intuitiva que podamos, ya que la finalidad última del proyecto son instituciones públicas. Debemos poder recibir datos de los dispositivos y enviar datos a dichos dispositivos. Con los datos recibidos posteriormente realizaremos estadísticas de consumo, los cuales mostraremos por medio de gráficas (siguiente versión del proyecto).

Nuestro proyecto lo podemos dividir en dos grandes etapas. Una primera etapa más teórica en la que no conocíamos la existencia del proyecto anterior. En la segunda etapa ya tomamos un enfoque más práctico y real.



1ª Etapa:

Se deseaba construir un interfaz gráfico en el que poder navegar por todas las habitaciones y ver en cada momento la temperatura de esa habitación y su temperatura objetivo. Además se quería tener un diagrama de sectores en el que en función del color (verde o rojo) nos avisase de diferentes alarmas. Verde significaría que las temperaturas registradas son las correctas y rojo significaría que las mediciones realizadas en algún momento del día se han pasado de los límites establecidos por el usuario. Además siempre se ha intentado tener un enfoque hacia el mundo táctil, con lo que hemos tenido que crear botones de un tamaño más grande de lo habitual.

También se trabajó con programas que nos ofrecían la capacidad de hacer zoom (debido al enfoque táctil comentado anteriormente). En esta parte del proyecto el tema de bases de datos quedaba un poco más al margen ya que nos centrábamos en la parte visual del proyecto

2ª Etapa:

Esta etapa surgió tras el interés de una empresa por el proyecto, esto supuso un cambio radical en el enfoque, puesto que ahora ya no primaba lo estético y visual, ahora lo más importante era la funcionalidad. Se creó una base de datos acorde a las necesidades del cliente y nos centramos en crear un interfaz gráfico amigable para el usuario en el que poder acceder y modificar todos los datos de nuestra base de datos.

En esta segunda etapa podemos dividir nuestro proyecto en 2 grandes módulos. Uno será el programa que se encargará de comunicarse con el comunicador (recibir datos del comunicador y enviarle datos al comunicador) valga la redundancia de nombres. Por otro lado tendremos el módulo principal en el que trataremos todos los temas relacionados con la base de datos. Aquí podremos desde dar de alta un radiador hasta asignar una programación a un radiador.



3.3 Objetivos

Los objetivos que se pretenden alcanzar en este proyecto se pueden resumir en los siguientes puntos:

- *Estudio de las herramientas, lenguajes y programas a utilizar en el diseño de la aplicación:*

Se realizará un estudio de las ventajas e inconvenientes de las diferentes opciones tratadas y se elegirá la que mas se ajuste a nuestro modelo, posteriormente esas decisiones se comunicarán al cliente, ya que este será el usuario final y es el que toma las decisiones.

Una vez el usuario final da el visto bueno a las ideas planteadas, se procederá a implantarlas en nuestro proyecto. Tenemos que tener en cuenta que nuestro proyecto es un proyecto abierto, es decir, tras la presentación del proyecto se seguirá desarrollando, así que tenemos que tener flexibilidad a los cambios ya que en un futuro puede que se realicen cambios. Primero nos centraremos en la funcionalidad y luego pasaremos a la parte estética.

Además debemos realizar las pruebas pertinentes con el otro proyecto con el que tiene que interactuar.

- *Crear documentación de apoyo para la aplicación.*

Este punto es muy importante puesto que el proyecto ha tenido una duración muy extensa y el programa esta destinado a usuarios con no necesarios conocimientos en informática.



3.4 Fases del Proyecto

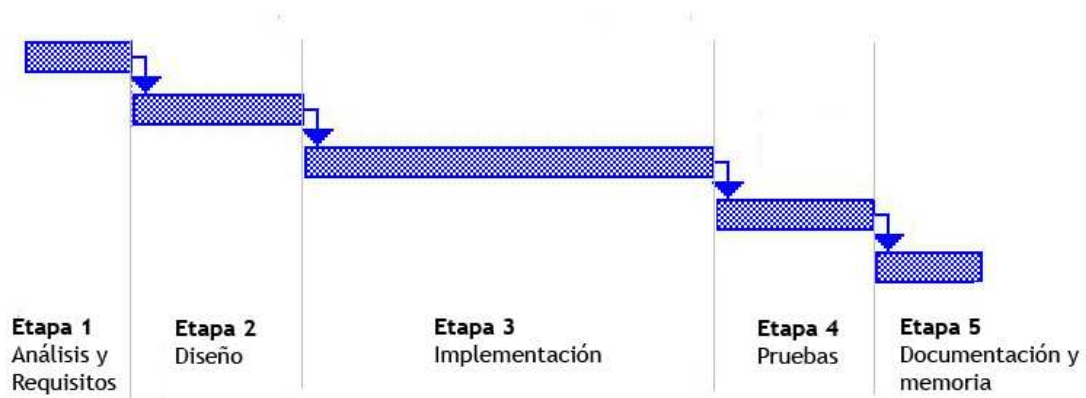
En esta parte de la memoria vamos a definir las diferentes fases por las que hemos pasado a lo largo del desarrollo de nuestro proyecto.

1. Formación y adaptación al lenguaje de programación
2. Análisis requisitos
3. Diseño del producto
4. Implementación
5. Pruebas
6. Documentación



3.5 Planificación

Ahora vamos a describir de forma mas detallada cada uno de los puntos citados anteriormente (3.4 Fases del Proyecto)



Análisis y Requisitos

Primero realizaremos un análisis de los diferentes lenguajes con los que podemos realizar nuestro proyecto. Al final nos hemos decantado por la utilización de Silverlight porque es un lenguaje orientado a la Web. Como todo el tema de trabajo en la Web, trabajo en la nube está muy de moda nos ha ayudado a tomar la decisión.

Tras la decisión del lenguaje, pasamos a una etapa de formación, ya que es un lenguaje con el que no he trabajado nunca. Este lenguaje lo podemos dividir en 2 partes, el lenguaje utilizado para el diseño (XAML) y el code-behind (C#). La adaptación no fue muy difícil debido a que C# es un lenguaje orientado a objetos y ya tenía una base de Java.



Además del aprendizaje del lenguaje tenemos que familiarizarnos con el Visual Studio 2010(programa con el que realizamos el desarrollo del programa).

En cuanto a los requisitos, se realizó una reunión al principio del proyecto con la empresa interesada en el producto y se acordaron los requisitos del sistema. Como el producto se va a seguir realizando a posterior de la presentación del proyecto teníamos que tener una mentalidad en la que el sistema tenía que ser flexible a futuros cambios.

Diseño

Como la empresa con la que estamos trabajando acababa de crear una nueva rama dedicada a las válvulas no disponíamos de base de datos, por lo que tuvimos que realizar nosotros el diseño ciñéndonos a los requisitos y necesidades que nos habían comentado en la primera reunión.

Tras varios prototipos de bases de datos conseguimos acertar en el diseño. Nuestro principal objetivo era poder guardar todos los datos del usuario y tener un fácil acceso a esos datos.

Implementación

Esta es la fase del proyecto en la que mas tiempo hemos invertido ya que estábamos trabajando con un lenguaje de programación nuevo y que además no lleva mucho tiempo en el mercado, con lo que hay bastantes temas que aun no se habían solucionado. Además la documentación ofrecida por Microsoft no era lo suficientemente clara.

Otro punto que ha alargado esta fase fue el largo periodo de tiempo entre reunión y reunión con los clientes, en cada reunión que se efectuaba se tenían que realizar cambios bastante grandes que nos llevaban gran cantidad tiempo realizarlos.

Pruebas

Esta fase del proyecto ha tenido mas importancia en el módulo que se encarga de enviar datos al comunicador, puesto que se podían dar muchas excepciones y tenemos que probar cada uno de esos casos, para que en un futuro no nos de problemas.



En la parte de las programaciones también hemos tenido que realizar muchas pruebas y ver que ocurría si se solapaban fechas, asignaciones múltiples...

Documentación

Como ya hemos mencionado en partes anteriores de la memoria, el usuario final de este producto va a ser una persona con no necesarios conocimientos de informática, debido a este tendremos que realizar un pequeño manual de usuario en el que explicar como realizar cada una de las acciones que nos ofrece el software creado.

Este manual se desarrollará en último lugar porque hemos intentado desarrollar el producto lo más rápido posible para una temprana incorporación al mercado.



4. Desarrollo

4.1 Adaptación a la empresa

La adaptación a la empresa fue rápida y sencilla, ya que es una empresa que no se dedica al sector de la informática y delegaban mucho en mí y mi tutor. Una de las principales bases de la empresa era la confianza. Periódicamente se realizaban reuniones con mi jefe D.Patxi Tornaría para mostrarle los progresos. Estas reuniones se realizaban porque mi puesto de trabajo no estaba en el edificio de Lator S.L. Todo el proyecto lo he realizado en el Centro de I+D en Electrónica y Comunicaciones Jerónimo de Ayanz, ya que mi empresa subcontrata a la universidad para investigación y desarrollo. Además mi proyecto tiene que interactuar con otro y la persona que estaba realizando ese proyecto también se encontraba ubicada en el edificio anteriormente mencionado.

En este edificio se llevan a cabo todos los proyectos de I+D de la universidad, y puesto que nuestro proyecto es nuevo en el sector, formamos parte del equipo de desarrollo del edificio Jeronimo de Ayanz



4.2 Estudio de alternativas

Ya desde el comienzo del proyecto se tenía la intención de usar silverlight, porque era un lenguaje que ya llevaba un tiempo en el mercado y había ido sacando diferentes versiones, por lo que nosotros dedujimos que se había mejorado, además es un lenguaje de Microsoft por lo que nos aseguraba el llegar a casi todos los equipos del mercado.

Pero antes de tomar la decisión final de realizar el proyecto en silverlight se realizó un estudio de otros lenguajes con los que poder crear aplicaciones Web, ya que esta era nuestra finalidad, poder crear una aplicación Web.

Ahora vamos a exponer brevemente los diferentes lenguajes que nos ofrecen unas características similares al lenguaje que hemos usado.

Adobe Flash:

Adobe Flash Professional es el nombre o marca comercial oficial que recibe uno de los programas más populares de la casa Adobe. Se trata de una aplicación de creación y manipulación de gráficos vectoriales con posibilidades de manejo de código mediante el lenguaje ActionScript en forma de estudio de animación que trabaja sobre "fotogramas" y está destinado a la producción y integración de contenido interactivo para las diferentes audiencias alrededor del mundo sin importar la plataforma.

Es actualmente desarrollado y comercializado por Adobe Systems Incorporated y forma parte de la familia Adobe Creative Suite, su distribución viene en diferentes presentaciones, que van desde su forma individual hasta como parte de un paquete siendo estos: Adobe Creative Suite Design Premium, Adobe Creative Suite Web Premium y Web Standard, Adobe Creative Suite Production Studio Premium y Adobe Creative Suite Master Collection. Su uso en las diferentes animaciones publicitarias, de reproducción de vídeos (como ocurre en YouTube) y otros medios interactivos que se presentan en casi todas las páginas Web del mundo le han dado la fama a éste programa dándoles el nombre de "animaciones Flash" a los contenidos creados con éste.



Diferencia con otros lenguajes:

La diferencia es que Adobe Flash utiliza las imágenes y sonidos "fotogramas" para crear animaciones 2D además es más fácil. Son utilizados en páginas Web y sitios Web multimedia. Estos pueden ser reproducidos por un reproductor Flash que también sirve para hacer multimedia como para crear animaciones, multimedia, juegos, etc.

Por otra parte, la animación es estilo de dar la sensación de movimiento a dibujos o imágenes tomadas a objetos reales y actores con el más minúsculo movimiento para crear una animación. El trabajo de crear imágenes en 3D sería mas tedioso.

Flex

Adobe Flex (hasta 2005 Macromedia Flex) es un término que agrupa una serie de tecnologías publicadas desde Marzo de 2004 por Macromedia para dar soporte al despliegue y desarrollo de Aplicaciones Enriquecidas de Internet, basadas en su plataforma propietaria Flash.

Los programadores tradicionales de aplicaciones ven como un desafío adaptar la metáfora de la animación sobre la plataforma con la cual fue originalmente construido Flash. Flex minimiza elegantemente este problema proviniendo un flujo de trabajo y un modelo de programación que es familiar a los desarrolladores de aplicaciones.

Flex fue inicialmente liberado como una aplicación de la J2EE o biblioteca de etiquetas JSP que compilaba el lenguaje de marcas Flex (MXML) y ejecutaba mediante ActionScript aplicaciones Flash (archivos SWF binarios). Versiones posteriores de Flex soportan la creación de archivos estáticos que son compilados, y que pueden ser distribuidos en línea sin la necesidad de tener una licencia de servidor.



El objetivo de Flex es permitir a los desarrolladores de aplicaciones Web construir rápida y fácilmente Rich Internet Applications, también llamadas RIAs. En un modelo multi-capa, las aplicaciones Flex son el nivel de presentación.

Flex pone en relieve el desarrollo de Interfaces gráficas de usuario usando un lenguaje XML llamado MXML. Flex tiene varios componentes y características que aportan funcionalidades tales como Servicios Web, objetos remotos, arrastrar y soltar, columnas ordenables, gráficas, efectos de animación y otras interacciones simples. El cliente solo carga la aplicación una vez, mejorando así el flujo de datos frente a aplicaciones basadas en HTML (PHP, ASP, JSP, CFMX), las cuales requieren de ejecutar plantillas en el servidor para cada acción. El lenguaje y la estructura de archivos de Flex buscan el desacoplamiento de la lógica y el diseño.

El servidor Flex también actúa como un gateway permitiendo al cliente comunicarse con servicios Web XML y objetos remotos (tales como Coldfusion CFCs, clases Java, y cualquiera que soporte el formato de mensajes de acciones).

JavaFX

JavaFX es una familia de productos y tecnologías de Sun Microsystems, adquirida por Oracle Corporation, para la creación de Rich Internet Applications (RIAs), esto es, aplicaciones Web que tienen las características y capacidades de aplicaciones de escritorio, incluyendo aplicaciones multimedia interactivas. Las tecnologías incluidas bajo la denominación JavaFX son JavaFX Script y JavaFX Mobile, aunque hay más productos JavaFX planeados.

Las aplicaciones JavaFX pueden ser ejecutadas en una amplia variedad de dispositivos. En su última versión (JavaFX 1.3, abril 2010) permite crear aplicaciones de escritorio, para celulares, la Web, TV, consolas de videojuegos, reproductores Blu-ray, entre otras plataformas planeadas. Para el desarrollo de aplicaciones JavaFX un lenguaje declarativo, tipado llamado JavaFX Script, además puede integrarse código Java en programas JavaFX. JavaFX es compilado a código Java, por lo que las



aplicaciones JavaFX pueden ser ejecutadas en computadores con la máquina virtual de Java instalada (JRE), o celulares corriendo Java ME.

JavaFX fue anunciado en la conferencia de desarrolladores JavaOne en mayo de 2007 y liberado en diciembre de 2008.

Ajax

Ajax, acrónimo de *Asynchronous JavaScript And XML* (JavaScript asíncrono y XML), es una técnica de desarrollo Web para crear aplicaciones interactivas o RIA (*Rich Internet Applications*). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se requieren al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. JavaScript es el lenguaje interpretado (scripting language) en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante *XMLHttpRequest*, objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML.

Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores dado que está basado en estándares abiertos como JavaScript y Document Object Model (DOM).

Inconvenientes:

- Las páginas con AJAX son más difíciles de desarrollar que las páginas estáticas.
- Las páginas creadas dinámicamente mediante peticiones sucesivas AJAX, no son registradas de forma automática en el historial del navegador, así que



- haciendo clic en el botón de "volver" del navegador, el usuario no será devuelto a un estado anterior de la página, en cambio puede volver a la última página que visitó. Soluciones incluyen el uso de IFrames invisible para desencadenar cambios en el historial del navegador y el cambio de la porción de anclaje de la dirección (después de un #).
- Los motores de búsquedas no entienden JavaScript. La información en la página dinámica no se almacena en los registros del buscador.
- Hay problemas usando Ajax entre nombres de dominios. Eso es una función de seguridad.
- El sitio con Ajax usa más recursos en el servidor. Recomendación: sólo usar las peticiones necesarias en Ajax, no desarrollar todo el sitio en AJAX. Con esto garantizamos menos recursos del servidor.
- Es posible que páginas con Ajax no puedan funcionar en teléfonos móviles, PDA u otros aparatos. Ajax no es compatible con todos los software para ciegos u otras discapacidades.

OpenLaszlo

El nombre Laszlo tiene origen Húngaro. Éste proyecto fue llamado así por el gato de Peter Andrea, un diseñador gráfico y co-fundador de Laszlo Systems. El gato había sido nombrado en honor del artista húngaro del constructivismo, pintor y fotógrafo László Moholy-Nagy.

OpenLaszlo es una plataforma código abierto para el desarrollo y distribución de RIA, Rich Internet Applications (Aplicaciones Ricas de Internet). Ha sido publicado bajo la licencia Common Public License, certificada por la Open Source Initiative.

La plataforma OpenLaszlo consiste en el lenguaje de programación LZX y el Servidor OpenLaszlo.



Las aplicaciones de Laszlo pueden ser desplegadas como tradicionales Java Servlets , que se compilan y regresan al navegador de manera dinámica. Este método requiere que en el servidor Web se ejecute el Servidor OpenLaszlo.

Alternativamente, las aplicaciones Laszlo pueden compilarse a partir de LZX en un Archivo binario SWF, y cargarse de manera estática en una página Web existente. Este método se conoce como despliegue SOLO.

HTML5

HTML 5 (*HyperText Markup Language*, versión 5) es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML. HTML 5 especifica dos variantes de sintaxis para HTML: un «clásico» HTML (text/html), la variante conocida como *HTML5* y una variante XHTML conocida como sintaxis *XHTML5* que deberá ser servida como XML (XHTML) (application/xhtml+xml). Esta es la primera vez que HTML y XHTML se han desarrollado en paralelo.

Todavía se encuentra en modo experimental, lo cual indica la misma W3C, aunque ya es usado por múltiples desarrolladores Web por sus avances, mejoras y ventajas.

Al no ser reconocido en viejas versiones de navegadores por sus nuevas etiquetas, se le recomienda al usuario común actualizar a la versión más actual, para poder disfrutar de todo el potencial que trae HTML 5. El desarrollo de este código es regulado por el Consorcio W3C.

Las principales novedades de este lenguaje son:

- Incorpora etiquetas (canvas 2D y 3D, audio, video) con codecs para mostrar los contenidos multimedia. Actualmente hay una lucha entre imponer codecs libres (WebM + VP8) o privativos (H.264/MPEG-4 AVC).



- Etiquetas para manejar grandes conjuntos de datos: Datagrid, Details, menu y Command. Permiten generar tablas dinámicas que pueden filtrar, ordenar y ocultar contenido en cliente.
- Mejoras en los formularios. Nuevos tipos de datos (eMail, number, url, datetime) y facilidades para validar el contenido sin Javascript.
- Visores: MathML (fórmulas matemáticas) y SVG (gráficos vectoriales). En general se deja abierto a poder interpretar otros lenguajes XML.
- Drag & Drop. Nueva funcionalidad para arrastrar objetos como imágenes.



4.3 Formación

En este apartado se explicarán las distintas tecnologías y herramientas que se han utilizado para el desarrollo de la aplicación. La decisión final de programar en Silverlight fue debido a que se intentó explorar ramas nuevas de programación ya que no se había usado anteriormente.

Por ello, como lenguaje de desarrollo Web se utilizará Microsoft Silverlight empleando, WCF Ria Services, y para el codebehind se utilizará el lenguaje C#. Como gestor de base de datos SQLserver y por último para acceder a los datos se utilizará ADO .Net. A continuación, se podrá encontrar una descripción de cada una de las herramientas y tecnologías utilizadas

Code-behind

Es una metodología de trabajo que nos ofrece silverlight. Los nombres de los archivos del code-behind van a ser del tipo *ejemplo.xaml.cs*, ya que está basado en la página xaml. En este código llamaremos a los constructores de la página.

Nosotros definiremos todos los eventos en el xaml, pero la definición de todos esos eventos los implementaremos en C# (lenguaje que utiliza el code-behind)

C#

C# (pronunciado *si sharp* en inglés) es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA (ECMA-334) e ISO (ISO/IEC 23270). C# es uno de los lenguajes de programación diseñados para la infraestructura de lenguaje común.

Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET, similar al de Java, aunque incluye mejoras derivadas de otros lenguajes.



El nombre C Sharp fue inspirado por la notación musical, donde # (sostenido, en inglés *sharp*) indica que la nota (*C* es la nota do en inglés) es un semitono más alta, sugiriendo que C# es superior a C/C++. Además, el signo '#' viene de cuatro '+' pegados.

Aunque C# forma parte de la plataforma .NET, ésta es una API, mientras que C# es un lenguaje de programación independiente diseñado para generar programas sobre dicha plataforma. Ya existe un compilador implementado que provee el marco Mono - DotGNU, el cual genera programas para distintas plataformas como Windows, Unix y GNU/Linux.

Ventajas del lenguaje:

- Lenguaje de programación orientado a objetos simple, moderno y de propósito general.
- Inclusión de principios de ingeniería de software tales como revisión estricta de los tipos de datos, revisión de límites de vectores, detección de intentos de usar variables no inicializadas, y recolección de basura automática.
- Capacidad para desarrollar componentes de software que se puedan usar en ambientes distribuidos.
- Portabilidad del código fuente.
- Fácil migración del programador al nuevo lenguaje, especialmente para programadores familiarizados con C, C++ y Java.
- Soporte para internacionalización.
- Adecuación para escribir aplicaciones de cualquier tamaño: desde las más grandes y sofisticadas como sistemas operativos hasta las más pequeñas funciones.
- Aplicaciones económicas en cuanto a memoria y procesado.



RIA: Rich Internet Applications

Las *rich Internet applications*, o *RIA* (en español "aplicaciones de Internet enriquecidas"), son aplicaciones Web que tienen la mayoría de las características de las aplicaciones de escritorio tradicionales. Estas aplicaciones utilizan un navegador Web estandarizado para ejecutarse y por medio de complementos o mediante una máquina virtual se agregan las características adicionales.

Las RIA surgen como una combinación de las ventajas que ofrecen las aplicaciones Web y las aplicaciones tradicionales. Buscan mejorar la experiencia del usuario.

Normalmente en las aplicaciones Web, hay una recarga continua de páginas cada vez que el usuario pulsa sobre un enlace. De esta forma se produce un tráfico muy alto entre el cliente y el servidor, llegando muchas veces a recargar la misma página con un cambio mínimo.

En los entornos RIA, en cambio, no se producen recargas de página, ya que desde el principio se carga toda la aplicación, y sólo se produce comunicación con el servidor cuando se necesitan datos externos como datos de una base de datos o de otros ficheros externos.

Las aplicaciones RIA son un nuevo tipo de aplicaciones con más ventajas que las tradicionales aplicaciones Web. Esta surge como una combinación de las ventajas que ofrecen las aplicaciones Web y las aplicaciones tradicionales.

Otra de las desventajas de las tradicionales aplicaciones Web es la poca capacidad multimedia que posee. Para ver un vídeo es necesario usar un programa externo para su reproducción.

Las capacidades multimedia son totales gracias a que estos entornos tienen reproductores internos y no hace falta ningún reproductor del sistema operativo del usuario.

Hay muchas herramientas para la creación de entornos RIA. Entre estas se puede mencionar las plataformas Adobe Flash, Adobe Flex y Adobe AIR de Adobe, AJAX,



OpenLaszlo, Silverlight de Microsoft, JavaFX Script de Sun Microsystems, Bindows, GWT, JavascriptMVC y Javascript.

Generalmente se tiene una aplicación cliente “stateful” y una capa de servicios separada. Las RIA se apoya más sobre un desarrollo “cliente-servidor” en vez de un desarrollo Web tradicional, en donde el estado se mantiene en el servidor en sesiones. El cliente sabe acerca de sí mismo y el tipo de datos que esta solicitando y únicamente solicita los datos que necesita sin ninguna otra información.

Cliente

Se maneja la interacción entre el usuario y la interfaz de usuario, el usuario invoca comandos, actualiza vistas y carga datos. Aquí se mantiene el estado de la aplicación, se manejan todas las peticiones de datos hacia el servidor y se controla como se presentan los datos.

Servidor

Aquí se manejan y se procesan todas las peticiones de la aplicación cliente y delega las acciones en el servidor, estas pueden ser, guardar datos en la base de datos, actualizar los archivos del sistema, retornar datos al servidor, o algún tipo de proceso analítico. Determina y le da formato a los datos que son retornados al cliente.

Silverlight

Microsoft Silverlight es un complemento de Microsoft que nos permite desarrollar aplicaciones enriquecidas para la Web. Silverlight funciona sobre varias plataformas y múltiples exploradores y proporciona una nueva generación de experiencias de usuario basadas en .NET. Tras una descarga que se instala en segundos, Silverlight posibilita una nueva plataforma rica, segura y escalable.

Silverlight ofrece un modelo de programación flexible y coherente compatible con lenguajes .NET como Visual Basic, C# y F#, y otros como AJAX, Python, Ruby y que además se integra con las aplicaciones Web existentes. Silverlight supone una nueva



forma de aprovechar los elementos multimedia en los principales navegadores entre los que se incluyen Firefox, Safari e Internet Explorer tanto en MacOS como en Windows.

Experiencias de usuario atractivas para diferentes plataformas

- Proporciona experiencias multimedia enriquecidas (RIA) para la Web que incorporan vídeo, animaciones, interactividad e interfaces de usuario sensacionales.
- Pequeña instalación gracias a un complemento de unos 5Mb, el cual es fácil de instalar y que funciona en los principales navegadores.
- Experiencias coherentes en Mac y Windows sin requisitos adicionales de instalación.
- Cree experiencias Web más completas y atractivas que aprovechan al máximo la capacidad del cliente para obtener el mejor rendimiento.
- Gráficos vectoriales, archivos multimedia, texto, animación y capas superpuestas permiten la integración perfecta de gráficos y efectos en cualquier aplicación Web existente.
- Mejore las aplicaciones existentes con unos gráficos y archivos multimedia más ricos, y mejore su rendimiento y capacidades con Silverlight.

Un modelo de programación flexible con herramientas de colaboración

- Basadas en .NET Framework, Silverlight permite que los desarrolladores y diseñadores usen de una manera fácil las habilidades y herramientas existentes para proporcionar experiencias de medios y RIA para la Web.
- La integración sencilla con las tecnologías y los activos Web existentes significa que Silverlight funciona con cualquier plataforma o tecnología Web back-end. Silverlight se integra con su infraestructura y aplicaciones existentes, que incluyen Apache, PHP, así como JavaScript y XHTML en el cliente.
- Diferentes opciones de lenguajes de desarrollo que incluyen C#, Visual Basic.NET y F# además de JavaScript, Ruby, Python, etc.



-Herramientas de funcionalidad específica tanto para diseñadores como para desarrollador es que aprovechan los estándares Web y la amplia variedad de características de la plataforma de aplicaciones .NET de Microsoft.

-Expression Studio (diseñadores) para crear interfaces de usuario interactivas y experiencias de medios enriquecidas, preparar medios para su codificación y distribución, y crear sitios compatibles con estándares W3C mediante los nuevos XHTML, XML, XSLT, CSS y ASP.NET.

-DeepZoom Composer (diseñadores) es un programa que nos permite crear multiescale imagen, este programa coge una foto y la subdivide en muchas partes así luego a la hora de cargarla podemos hacer zoom de una manera mas rápida sin tener que esperar que se cargue toda la imagen.

-Visual Studio (desarrolladores) para el desarrollo de código de cliente y servidor con la característica Intellisense completa, depuración eficaz en varias plataformas, compatibilidad con lenguaje enriquecido y mucho más.

- Modelo de presentación coherente con XAML, el lenguaje de presentación declarativo usado en aplicaciones de Windows 7. Los controles, diseños visuales, archivos multimedia y otros elementos se pueden presentar con total fidelidad de diseño en las aplicaciones de Silverlight y de Windows.

- El modelo de control extensible facilita la adición de contenido y comportamientos enriquecidos mientras que permite la reutilización y el uso compartido del código.

Totalmente multimedia

- El formato de archivos multimedia unificado abarca desde contenidos HD hasta formatos para dispositivos móviles mediante Vídeo de Windows Media (WMV), la implementación de Microsoft del estándar SMPTE VC-1 de video, al igual que compatibilidad con audio WMA y MP3.

- Agregue gráficos vectoriales y capas compatibles con la integración de gráficos ampliables hasta cualquier tamaño, así como superposiciones con estilo de emisión televisiva para la adición de bandas y subtítulos (CC).



- Soluciones flexibles de inserción de anuncios con vídeos y animaciones que incluyen la capacidad de proporcionar vídeo con estilo de emisión televisiva o anuncios animados sin pérdidas de fidelidad visual ni calidad de movimiento.
- Amplio ecosistema de herramientas de multimedia, servidores y soluciones compatible con la plataforma Tecnologías de Windows Media.
- Herramienta de codificación eficaz para la publicación en vivo y a petición de experiencias de medios con Expression Media Encoder, que incluye la codificación con aceleración de hardware de WMV, VC-1, H.264, AAC y otros.

Servidores y servicios conectados a la Web

- Desarrolle fácilmente aplicaciones de Web híbridas e incorpore los servicios y datos de cualquier parte de la Web gracias a la compatibilidad de Silverlight con LINQ y LINQ a XML mientras se obtiene acceso a esos datos con protocolos comunes como JSON, RSS, POX y REST
- Aumento de la capacidad de detección(SEO) de contenidos RIA que se pueden indexar y buscar gracias al formato XAML basado en texto, que describe la interfaz y el contenido en una aplicación de Silverlight.

WCF RIA Services

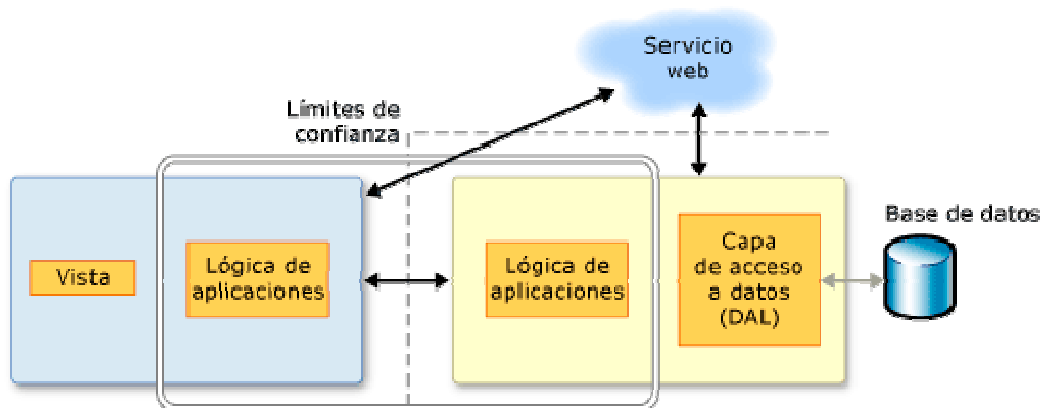
WCF RIA Services simplifica el desarrollo de soluciones de n niveles para aplicaciones de Internet enriquecidas (RIA), como las aplicaciones de Silverlight. Un problema común a la hora de desarrollar una solución de RIA de n niveles es la coordinación de la lógica de aplicación entre el nivel intermedio y el nivel de presentación.

Para crear la mejor experiencia de usuario, desea que el cliente de RIA Services conozca la lógica de aplicación que reside en el servidor pero no desea que se desarrolle ni se mantenga la lógica de aplicación tanto en el nivel de presentación como en el nivel intermedio. RIA Services soluciona este problema proporcionando componentes, herramientas y servicios de marco que ponen la lógica de aplicación del servidor a disposición del cliente de RIA Services sin que sea necesario duplicar manualmente la



lógica de programación. Puede crear un cliente de RIA Services que conozca las reglas de negocios y sepa que el cliente se actualiza automáticamente con la última lógica de nivel intermedio cada vez que se recompila la solución.

En la ilustración siguiente se muestra una versión simplificada de una aplicación de n niveles. RIA Services se centra en el cuadro entre el nivel de presentación y la capa de acceso a datos (DAL) para facilitar el desarrollo de n niveles con un cliente de RIA Services.



Integración de WCF

En RIA Services, puede exponer datos del proyecto de servidor en el proyecto de cliente agregando servicios de dominio. El marco de RIA Services implementa cada servicio de dominio como un servicio de Windows Communication Foundation (WCF). Por lo tanto, puede aplicar los conceptos que conoce de WCF Services a los servicios de dominio a la hora de personalizar la configuración

Servicios de dominio y sus orígenes de datos

La clase DomainService es la clase base para todas las clases que actúan como servicios de dominio. Para crear un servicio de dominio que enlaza a un objeto de datos personalizado, debe crear una clase que derive directamente de DomainService. Pero si tiene un servicio de dominio que enlaza un Entity Data Model de ADO.NET o que



expone un vínculo a una base de datos SQL, puede utilizar en su lugar clases abstractas especiales que derivan de `DomainService`.

- Para crear un servicio de dominio que enlace a un Entity Data Model dADO.NET, debe crear una clase que derive de `LinqToEntitiesDomainService`. RIA Services proporciona la clase `LinqToEntitiesDomainService`.

- Si desea crear un servicio de dominio que exponga clases de LINQ to SQL en la aplicación, debe crear una clase que derive de `LinqToSqlDomainService`. Esta clase se proporciona en el Kit de herramientas de RIA Services. Si desea crear un servicio de dominio que exponga clases de LINQ to SQL en la aplicación utilizando esta clase, debe descargar el Kit de herramientas de RIA Services .

Cuando se utiliza el cuadro de diálogo Agregar nueva clase de servicio de dominio para crear un servicio de dominio, el tipo de servicio de dominio que se crea está basado en las entidades que se exponen.

Una clase de servicio de dominio se debe marcar con el atributo `EnableClientAccessAttribute` para que el servicio esté disponible para el proyecto de cliente. El atributo `EnableClientAccessAttribute` se aplica automáticamente al servicio de dominio cuando se activa la casilla Habilitar acceso de cliente del cuadro de diálogo Agregar nueva clase de servicio de dominio. Cuando se aplica el atributo `EnableClientAccessAttribute` a un servicio de dominio, RIA Services genera las clases correspondientes para el proyecto de cliente. Por ejemplo, cuando se aplica el atributo `EnableClientAccessAttribute` a un servicio de dominio denominado `HRService` que expone una entidad denominada `Employee`, RIA Services genera un contexto de dominio denominado `HRContext` en el proyecto de cliente y una versión de cliente de la entidad `Employee`.



WCF y servicios de dominio

Como servicio de Windows Communication Foundation (WCF), el servicio de dominio se basa en los conceptos de WCF. El servicio de dominio conserva las funciones siguientes:

- Uso estándar de servicios de WCF
- Existencia de estructuras de modelos de programación de WCF, como contratos de operación, comportamientos de operación y comportamientos de servicio
- Capacidades de personalización de WCF estándar, como configuración de enlace, configuración de comportamiento e infraestructura de administración

El contexto de dominio se comunica con el servicio de dominio de RIA Services utilizando el elemento ChannelFactory de WCF para crear un canal y pasarle un contrato de servicio generado desde el servicio de dominio.

De forma predeterminada, solo el extremo binario está habilitado para servicios de dominio. Para utilizar el extremo binario no se necesita ninguna configuración adicional. Si desea utilizar otro extremo (como OData, JSON, SOAP o un host personalizado) debe registrar un generador de extremos en el archivo Web.config como se muestra a continuación:

Contiene los extremos que se admiten en RIA Services . El espacio de nombres Microsoft.ServiceModel.DomainServices.Hosting contiene los extremos que se admiten con el Kit de herramientas de RIA Services , como el elemento JsonEndpointFactory que se muestra en el ejemplo anterior. Para crear un extremo personalizado, debe crear una clase que derive de la clase DomainServiceEndpointFactory y reemplazar el método CreateEndpoints.



Operaciones de datos

Puede agregar métodos a un servicio de dominio que realicen la operación de datos que desee exponer. Por ejemplo, puede agregar métodos que realicen las siguientes operaciones:

- Query
- Update
- Insert
- Delete

También puede agregar operaciones más complicadas como:

- Invoke: para implementar operaciones que necesitan ejecutarse sin seguimiento ni aplazamiento. Este método se utiliza únicamente con datos que no sean de entidad y solo cuando no se puedan usar operaciones de consulta, actualización, inserción o eliminación en su lugar.

- Named Update: para implementar operaciones personalizadas que no pertenecen a operaciones de modificación simples.

Cuando se expone un servicio de dominio, se genera un objeto EntitySet en el contexto de dominio con propiedades que indican qué operaciones (inserción, actualización o eliminación) se permiten en el cliente. Las modificaciones de datos se ejecutan modificando la colección de entidades y llamando después al método SubmitChanges.

En casi todos los escenarios, se deben usar operaciones de consulta en lugar de operaciones de invocación para la carga de datos. Los métodos de consulta devuelven un solo objeto Entity, un objeto IQueryable<Entity> o un objeto IEnumerable<Entity>. Los métodos de consulta forman parte integral del patrón de datos admitido por DomainService en el nivel intermedio y por DomainContext en el cliente. El marco de RIA Services genera entidades en el proyecto de cliente solo para las entidades devueltas por los métodos de consulta de un elemento DomainService.

Las operaciones de invocación proporcionan un mecanismo fuera de banda para devolver datos que no son de entidad y ejecutar operaciones con efectos secundarios.



Para obtener más información sobre los efectos secundarios, vea la propiedad HasSideEffects. Las operaciones de invocación no son normalmente adecuadas para los métodos de consulta. Incluso cuando una operación de invocación devuelve una entidad, la entidad se genera para el proyecto de cliente únicamente si la devuelve un método de consulta.



4.4 Análisis de requisitos

4.4.1 Análisis de Casos de Uso

En esta etapa del proyecto se establecen las características de nuestra aplicación. Los requisitos se establecerán tras las primeras reuniones con la empresa, además se ha pedido la opinión a otros expertos en la materia. Se mantuvo contacto con D.Luis Orus Jefe de Sección de Mantenimiento de Centros y Dependencias del Gobierno de Navarra. En estas reuniones tratábamos de predecir todas las posibles situaciones que se pueden dar en un colegio. Una vez conocidas las situaciones básicas, pasábamos a tratar las excepciones más comunes y cada vez se afinaba mas en la excepción hasta llegar a casos muy concretos, de esta manera nos aseguramos en un futuro tener controladas el mayor número de situaciones.

A continuación pasaremos a detallar las acciones que puede llevar a cabo el usuario final del software que me hemos desarrollado:

- *Gestión de Radiadores*: El usuario podrá tanto dar de alta un radiador como borrarlo o modificarlo. Dentro de la gestión de los radiadores podrá dar de baja un radiador, la cual no tendrá los mismos efectos que borrar un radiador.

- *Gestión Características Radiador*: En esta pantalla el usuario podrá tener acceso a todos los datos de las diferentes partes del radiador:

- Gestión Angulo Válvula*: Desde esta pestaña podremos dar de alta, baja y modificar un ángulo para la válvula de un radiador.

- Gestión Diámetro Entrada*: Desde esta pestaña podremos dar de alta, baja y modificar el diámetro de entrada del radiador.

- Gestión Forma Válvula*: Desde esta pestaña podremos dar de alta, baja y modificar la forma de la válvula del radiador.

- Gestión Material Tubo*: El usuario podrá dar de alta, baja o modificar los diferentes materiales que pueden tener los tubos del radiador.



-Gestión Posición Carcasa: Desde esta pestaña podremos dar de alta, baja y modificar la posición de la carcasa del radiador.

-Gestión Posición Válvula: El usuario final tendrá la posibilidad de dar de alta, baja y modificar la posición de la válvula que tendrán los radiadores.

-Gestión Racor: El usuario podrá crear, borrar y modificar los diferentes racores que van a tener los radiadores.

-Gestión Ubicación Toma: Desde esta última pestaña podremos crear, borrar y modificar las diferentes ubicaciones de la toma que van a tener nuestros radiadores.

· *Gestión Localización*: Crearemos las diferentes localizaciones en las que estarán ubicados los radiadores de nuestro colegio.

· *Gestión Partes Localización*: En esta pantalla tendremos acceso a los diferentes edificios, plantas y salas en las que tendremos ubicados nuestros radiadores.

-Gestión de Edificios: Desde esta pestaña podremos crear, borrar y modificar los edificios en los que tendremos ubicados los radiadores.

-Gestión de Plantas: Desde esta pestaña podremos crear, borrar y modificar las plantas en las que tendremos situados los radiadores.

-Gestión de Salas: Desde esta pestaña podremos crear, borrar y modificar las diferentes salas en las que tendremos ubicados los radiadores.

· *Gestión de Grupos*: Podemos crear, borrar y modificar los diferentes grupos en los que tendremos asignados los radiadores.

· *Gestión de Asignaciones*: Desde esta página podremos llevar a cabo todas las asignaciones que puede tener un radiador. Podremos tener asignaciones de tipo, radiador a grupo y localización a radiador.

· *Gestión Asignaciones Radiador a Grupo*: Desde esta pantalla podremos asignar los radiadores a los grupos creados y también podremos quitar radiadores de grupos.

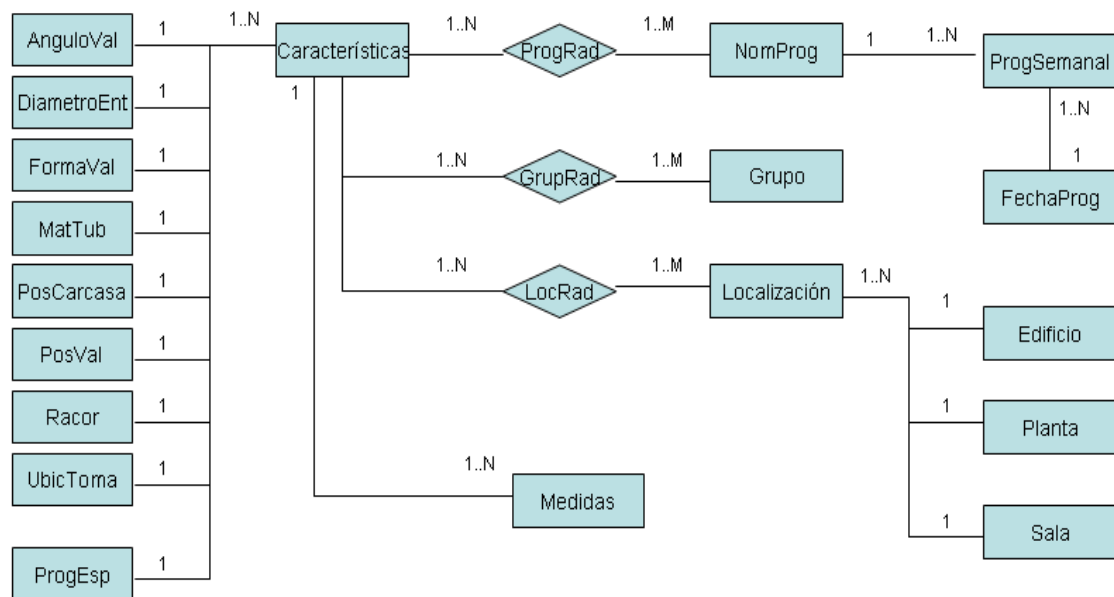


- *Gestión Asignaciones Localización a Radiador*: Al igual que en el caso de uso anterior podremos asignar y desasignar radiadores de localizaciones.
- *Gestión de Programaciones*: Desde esta página podremos crear, borrar y modificar programaciones, además también podremos asignar programaciones a los radiadores.
- *Gestión de Programaciones especiales*: Podremos crear, modificar y borrar las diferentes programaciones especiales. También tendremos acceso a la asignación de programaciones especiales los radiadores.
- *Gestión de medidas*: Desde aquí podremos visualizar las medidas reflejadas por los diferentes sensores de nuestro sistema. También podremos aplicar diferentes filtros.



4.4.2 Análisis Modelo Entidad-Relación

La base de datos se modeló desde 0 debido a que la empresa acaba de montar esta rama de trabajo. Teniendo en cuenta todos los requisitos y necesidades del cliente nos ha quedado el siguiente diagrama E-R:





- Las entidades del modelo son:

AnguloVal: Esta tabla existe por si misma y en ella tenemos guardados todos los ángulos que pueden tomar las válvulas de los radiadores de nuestra base de datos.

DiametroEnt: Esta tabla tiene existencia por si misma y en ella tenemos guardados todos los diámetros de entrada que pueden tomar las válvulas de los radiadores de nuestra base de datos.

FormaVal: Esta tabla existe por si misma y en ella tenemos guardadas todas las formas que pueden tomar las válvulas de los radiadores de nuestra base de datos.

MaterialTub: Esta tabla tiene existencia por si misma y en ella tenemos guardados todos los materiales de los tubos de los radiadores de nuestra base de datos.

PosCarcasa: Esta tabla existe por si misma y en ella tenemos guardadas todas las posiciones de las carcassas de los radiadores de nuestra base de datos.

PosVal: Esta tabla existe por si misma y en ella tenemos guardadas todas las posiciones de las válvulas de los radiadores de nuestra base de datos.

Racor: Esta tabla tiene existencia por si misma y en ella tenemos guardados todos los racores de los tubos de los radiadores de nuestra base de datos.

Ubicacion Toma: Esta tabla existe por si misma y en ella tenemos guardadas todas las ubicaciones de las tomas de las válvulas de los radiadores de nuestra base de datos.

ProgEsp: Esta tabla existe por si misma y en ella guardamos todas las programaciones especiales que tengamos definidas.

Caracteristicas: Esta tabla no puede existir si desapareciesen todas las tablas anteriores. Ya que es la unión de las claves primarias de las tablas anteriores, de esta manera guardamos todas las especificaciones de los radiadores.



Edificio: Tabla con existencia propia en la que guardamos en nombre de los tipos de edificios de nuestra base de datos

Planta: Tabla con existencia propia en la que guardamos las plantas que tiene el edificio sobre el que estamos implantando nuestro software

Sala: Esta tabla existe por si sola y guardamos los nombres de las salas de edificio en el que tenemos implantado nuestro software

Localización: Tabla sin existencia propia, ya que si desapareciesen las 3 tablas anteriores esta no tendría datos de los que alimentarse, aquí formamos localizaciones uniendo un atributo de cada una de las 3 tablas mencionadas anteriormente.

Grupo: Tabla con existencia propia en la que guardamos los grupos que tenemos definidos en la base de datos. Los grupos los crearemos para simplificar las asignaciones de las programaciones a los radiadores.

FechaProg: Tabla con existencia propia en la que tendremos guardados los intervalos de fechas que asignaremos posteriormente a las programaciones.

NomProg: Tabla en la que guardaremos el nombre de las programaciones que usaremos a lo largo del curso.

ProgSemanal: Tabla sin existencia propia ya que si desapareciesen las tablas TFechaProg o TNomProg, esta perdería el sentido, esta tabla toma la clave primaria de TFechaProg y la clave primaria de TNomProg. En esta tabla tenemos guardados los días de la semana con las temperaturas objetivo guardadas en intervalos de 30 min.

Medidas: Tabla en la que guardamos las medidas reflejadas en cada radiador, esta tabla no tiene existencia propia ya que si desapareciese la tabla TCaracteristicas no tendría ningún sentido tener, ya que tendríamos las medidas de algo sin saber quien las ha enviado.



- Las relaciones del modelo son:

- TAnguloVal-TCaracteristicas: esta relación es 1:N, ya que un ángulo puede estar en (0,N) radiadores y un radiador solo puede tener un ángulo de válvula.
 - TDiametroEnt-TCaracteristicas: esta relación es 1:N, un diámetro de entrada puede estar en (0,N) radiadores, pero un radiador solo puede tener un diámetro de entrada.
 - TFormaVal-TCaracteristicas: relación de 1:N, una forma de la válvula puede estar en (0,N) radiadores, pero un radiador solo puede tener una forma de válvula.
 - TMaterialTub-TCaracteristicas: relación de 1:N, un material de un tubo puede estar en (0,N) radiadores, pero un radiador solo puede tener el tubo de un material.
 - TPosCarcasa-TCaracteristicas: relación de 1:N, una posición de una carcasa puede estar en (0,N) radiadores, pero un radiador solo puede tener una posición de carcasa.
 - TPosVal-TCaracteristicas: relación de 1:N, una posición de la válvula puede estar en (0,N) radiadores, pero un radiador solo puede tener una posición de válvula.
 - TRacor- TCaracteristicas: relación de 1:N, un racor puede estar en (0,N) radiadores, pero un radiador solo puede tener un racor.
 - TUbicaciónToma- TCaracteristicas: relación de 1:N, una ubicación de toma puede estar en (0,N) radiadores, pero un radiador solo puede tener una ubicación de toma.
 - TProgEsp-TCaracteristicas: relación de 1:N, una programación especial puede estar en (0,N) radiadores, pero un radiador solo puede tener una programación especial.
- Localización:* es la unión de un edificio, planta y sala junto a un identificador autonumérico.
- TEedificio- TLocalizacion: relación de 1:N, un edificio puede estar en (0,N) localizaciones, pero una localización solo puede tener un edificio.



·TPlanta- TLocalizacion: relación de 1:N, una planta puede estar en (0,N) localizaciones, pero una localización solo puede tener una planta.

·TSala- TLocalizacion: relación de 1:N, una sala puede estar en (0,N) localizaciones, pero una localización solo puede tener una sala.

·TLocalización-TCaracteristicas: relación de N:M, ya que un radiador(modelo genérico de radiador) puede estar en (0,N) localizaciones y una localización puede tener (0,N) radiadores. De esta relación nace la tabla TLocalizCarac.

·Grupo-TCaracteristicas: relación de N:M, ya que un radiador(modelo genérico de radiador) puede estar en (0,N) grupo y un grupo puede tener (0,N) radiadores. De esta relación nace la tabla TGrupoCarac.

·TFechaProg- TProgSemanal: relación de 1:N, una fecha de programación puede estar en (0,N) programaciones semanales, pero una programación semanal solo puede tener una fecha.

·TNomProg- TProgSemanal: relación de 1:1, un nombre de programación puede estar en (0,1) programaciones semanales, y una programación semanal solo puede tener un nombre de programación.

· TNomProg-TCaracteristicas: relación N:M, debido a que una programación puede estar en 1 o en varios radiadores y un radiador puede tener 1 o varias programaciones. Podemos tener radiadores sin programación y programaciones sin radiador.

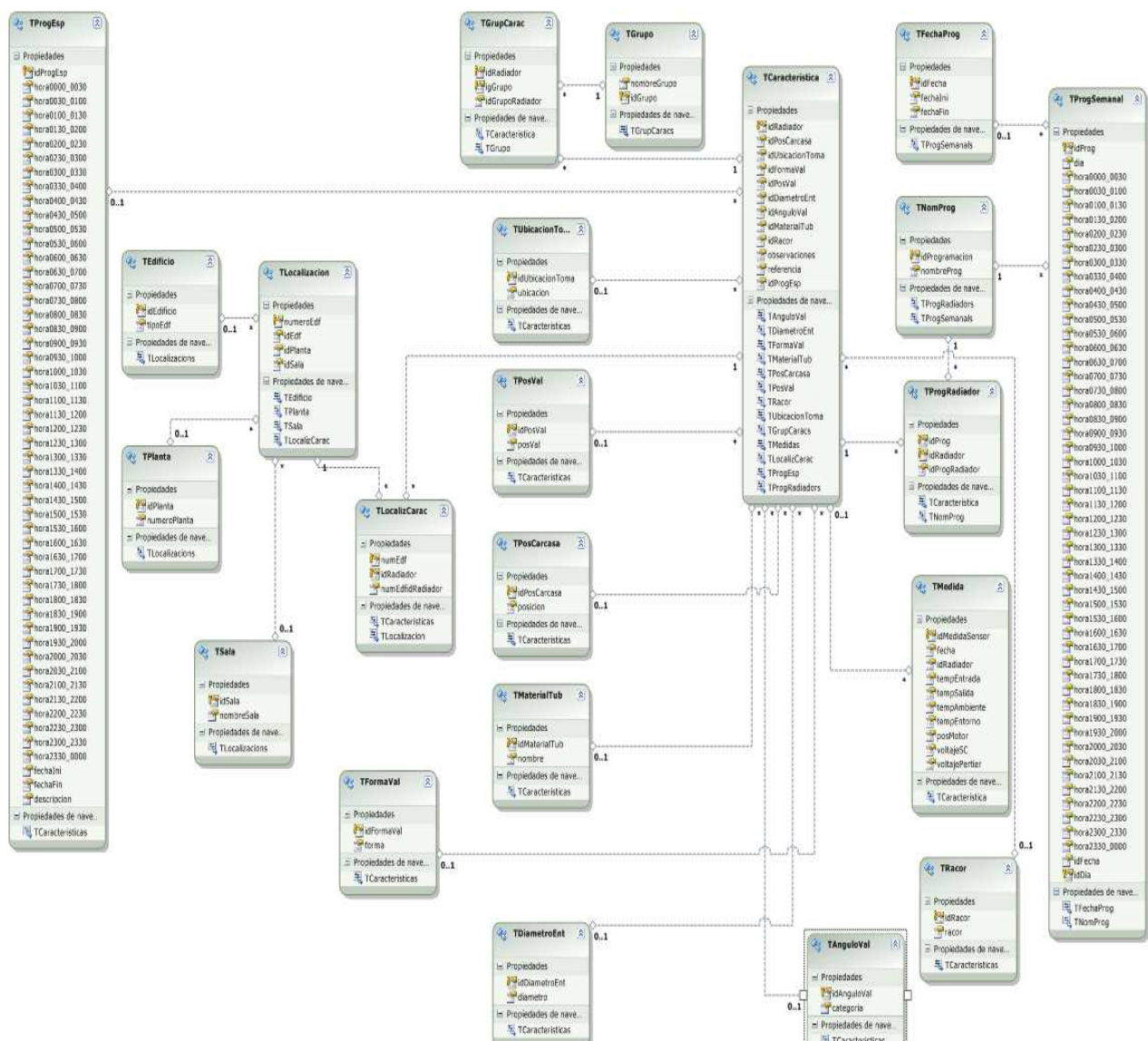
·TCaracteristicas-TMedidas: relación 1:N, ya que un radiador puede tener (0,N) medidas, pero una medida puede ser de un solo radiador



4.5 Diseño e implementación

4.5.1 Diseño modelo E-R

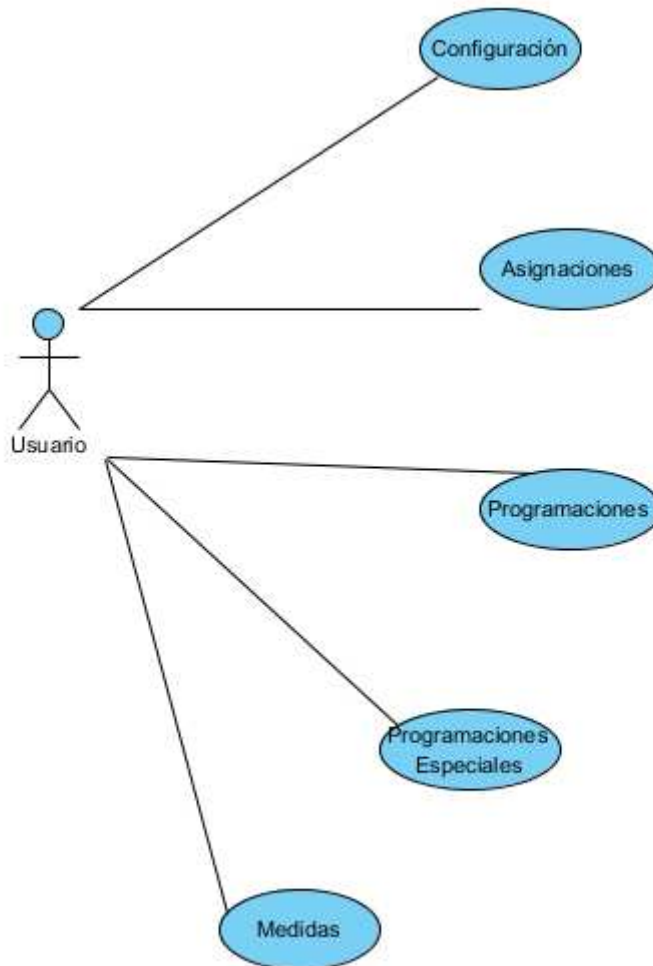
Una vez tenemos el diagrama E-R pasamos a modelar la base de datos en el visual Studio. Tras haber realizado el paso a tablas teniendo en cuenta todas las relaciones 1..N y N..M nos han quedado las siguientes tablas con los siguientes atributos.





4.5.2 Diseño Casos De Uso

Caso de uso pantalla Inicial

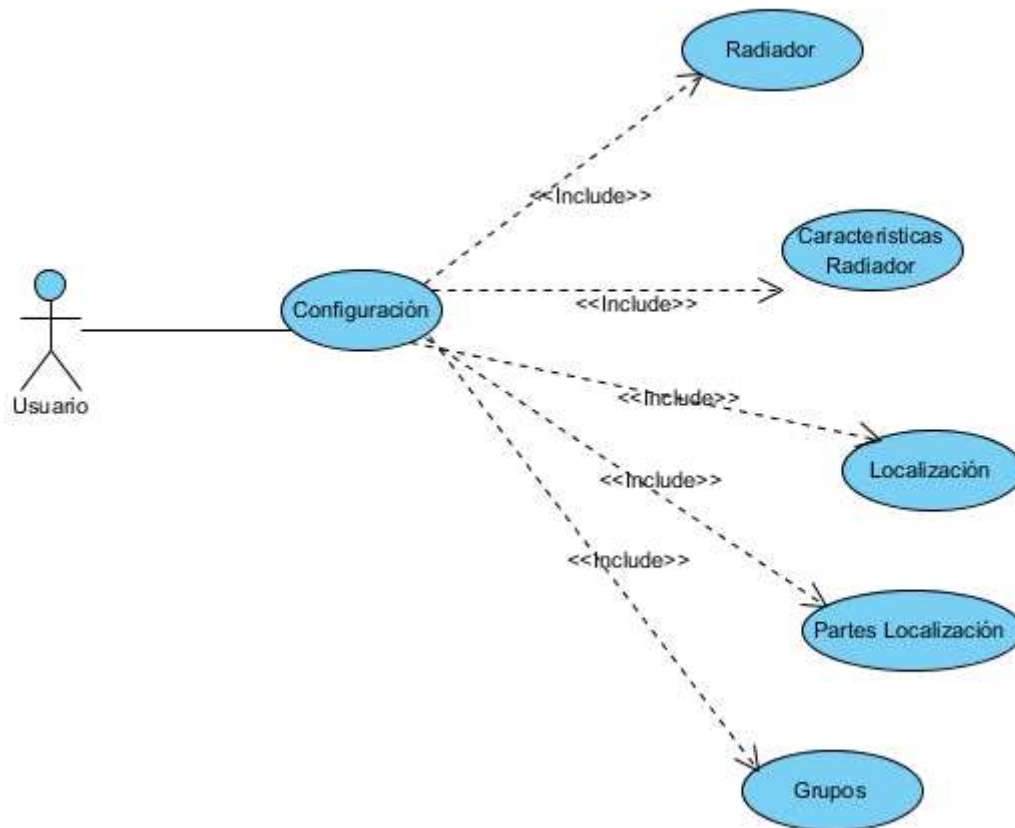


En este caso de uso podemos ver todas las diferentes opciones que tiene el usuario nada más arrancar el software. Por defecto arrancamos en el Home, desde esta página podemos navegar por todas las plantas y salas.

El usuario puede ir a la configuración del sistema, asignaciones, programaciones, programaciones especiales y visualizar las medidas registradas por los sensores



Caso de Uso Configuración



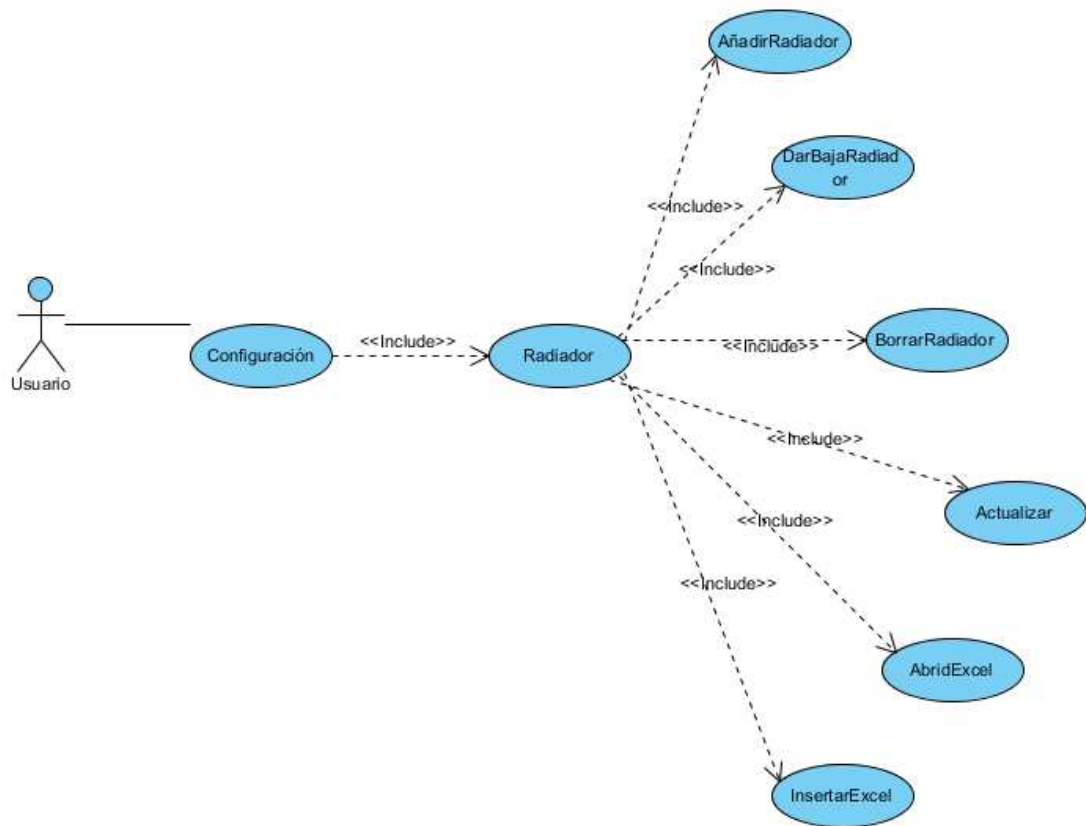
Una vez clicamos sobre la pestaña de configuración tendremos la posibilidad de realizar las siguientes acciones:

Podremos ir a la configuración de un radiador, también podremos ir acceder a la información de cada uno de los componentes de los radiadores.

Otra pestaña que nos encontraremos será la de Localización desde la cual obtendremos toda la información de las localizaciones, además de poder tener acceso a todos los componentes de las localizaciones. Por ultimo tendremos la opción de acceder a los datos de los grupos



Caso de Uso Gestión de Radiadores



Desde la configuración de radiadores podremos crear, borrar o actualizar los radiadores. Haremos una diferenciación entre dar de baja y borrar un radiador, dar de baja consistirá en borrarle las programaciones y quitarle los grupos a los que está asignado, pero mantendremos en la base de datos el radiador y sus medidas.

En el caso de borrar un radiador se borrará de forma completa, tanto las programaciones y grupos como las medidas y su aparición en la base de datos, antes de

ser borrado se comprobarán sus asignaciones y se mostrarán mensajes de advertencia por cada asignación(grupo, programación y medidas).



Debido a que la cantidad de radiadores va a ser alta, hemos buscado una forma alternativa de insertar los datos, esta forma va a ser cargar los datos desde un Excel.

Desde el Excel podremos además de cargar radiadores podremos crear localizaciones y asignar directamente la localización al radiador. En caso de no existir la localización en la base de datos crearemos una nueva, pero si ya existe no haremos nada. Lo mismo ocurrirá con las partes de la localización, si no existen se informará al usuario de la inexistencia de esa parte de la localización y en caso de “Aceptar” crearemos la localización del Excel, si no acepta, el radiador creado no tendrá localización y tendremos que realizar la asignación de forma manual.

De ahí aparece los botones AbrirExcel desde el cual buscaremos en nuestro ordenador el Excel seleccionado y InsertarExcel desde el cual se insertan los valores cargados a partir del botón anterior.

Para tratar con el Excel nos hemos basado en la librería XLSParsingLib con la que podemos leer hojas de Excel a partir de la versión 2007. Esta librería nos permite subir a nuestro servidor los datos de dicha hoja, luego nosotros ya nos encargaremos de tratar esos datos.

AbrirExcel: Al presionar sobre este botón pasaremos a buscar en nuestro ordenador el Excel que deseamos cargar. Un vez encontrado lo abriremos y los datos de ese Excel los guardaremos en un datagrid para visualizar desde nuestro programa si es realmente el Excel que estábamos buscando.

Leeremos la primera línea del Excel y de ahí obtendremos el título de las columnas del datagrid.

InsertarExcel: A partir de los datos que tenemos en el datagrid insertaremos esos valores en nuestra base de datos. Primero trataremos los datos de las localizaciones. Todas las consultas de existencia explicado anteriormente. Luego haremos una consulta en la base de datos a ver si la combinación de Edificio, Planta y Sala existe en nuestra



base de datos, si ya existe no hacemos nada pero si no existe la crearemos. En caso de no existir alguno de estos elementos por separado se creará automáticamente, al igual que la combinación Edificio, Planta, Sala. Debemos tener mucho cuidado con esto porque Silverlight trabaja de forma asíncrona y nosotros estamos insertando muchos valores a la vez. Este problema lo hemos solucionado guardando las localizaciones insertadas, así que ahora cada consulta antes de crear una nueva localización la hacemos sobre 2 tablas, sobre la de la base de datos y sobre la tabla que nos hemos creado de insertados.

Otra funcionalidad que nos ofrece el software es la posibilidad de crear grupos a partir de las iniciales de las salas. Por convenio con la empresa, todas las que empiecen por C, son del grupo comedor, por H hall... Al igual que con los edificios si no existe se creará antes de asignárselo al radiador.

Una vez tratados todos los datos de las localizaciones pasamos a tratar los datos de los radiadores. Haremos algo parecido a lo anterior pero más fácil porque el radiador tiene un número de serie que es único. Hacemos una consulta en la base de datos y si el radiador ya existe mostramos un mensaje en el que indicamos que el radiador X ya existe en la base de datos, si el radiador no existe lo insertamos en la base de datos.

Tras haber creado las localizaciones y los radiadores los ubicaremos en nuestro edificio.

Cada vez que insertamos un radiador nuevo llamamos a la función AsignaLoc(), esta función toma como argumentos una clase que hemos creado:

```
public class RadLoc
{
    public string numSerie;
    public int idEdf;
    public int idPlanta;
    public int idSala;
}
```

En esta clase guardamos el dato que nos identifica de manera única el radiador(no es el id de la base de datos) y los 3 campos que hacen única a la localización.



¿Por qué crear esta clase?

Porque los identificadores de los objetos TCaracterísticas y TLocalización se crean de forma automática justo cuando son insertados y nosotros tenemos que hacer muchas inserciones y hasta que no leemos todos los valores del datagrid no aplicamos los cambios. Para ese momento ya no sabríamos que radiadores se han insertado y cuales no.

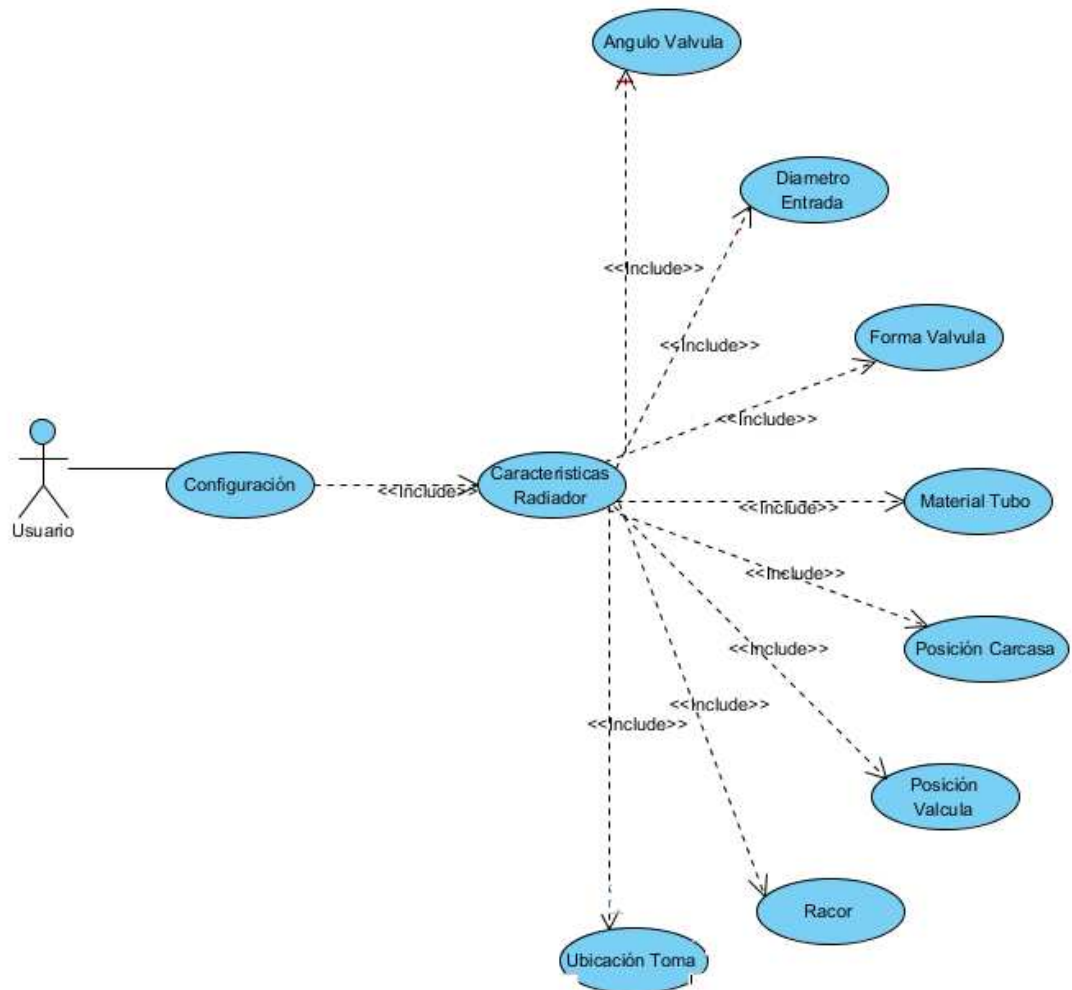
A partir de ese problema creamos la clase anterior en la que guardaremos en una lista de objetos “RadLoc” el radiador que se ha insertado y su localización.

Una vez se haya terminado el proceso de inserción ya tendremos los identificadores de la base de datos. Usando la clase anterior consultaremos en la BD dichos ids y procederemos a la asociación del radiador con su localización.

Más adelante desde el Excel se podrán hacer más cosas, esto se explicará en las líneas futuras del proyecto.



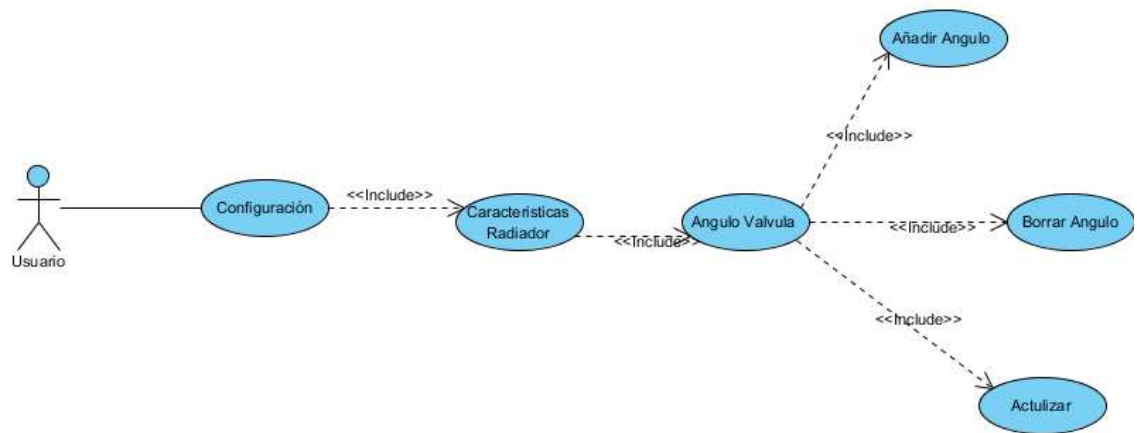
Caso de Uso Gestión de Características Radiador



En este caso de uso accederemos a otras sub-pestañas en las que tendremos el control de todos y cada uno de los componentes de nuestro radiador.



Caso de Uso Gestión Angulo Válvula



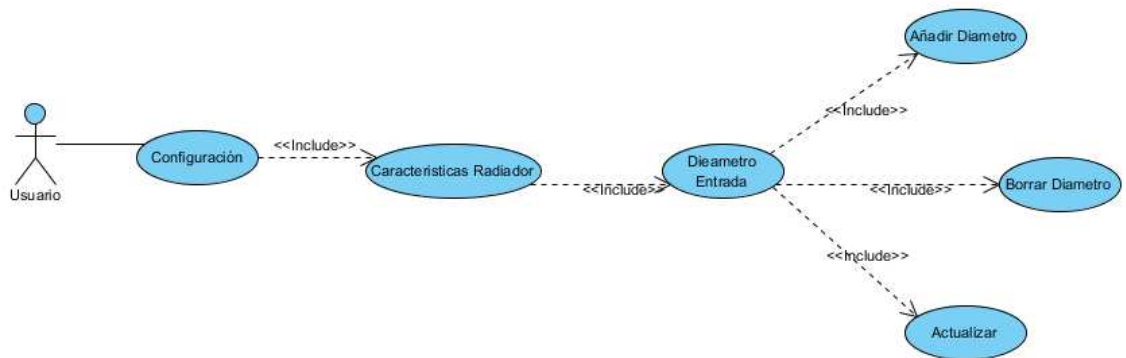
En este caso de uso podremos crear un ángulo nuevo, también podremos borrar el ángulo que seleccionemos y por último podremos modificar un ángulo, solo se nos permitirá cambiar la especificación del ángulo, nunca la clave primaria.

En el caso de borrar primero nos preguntará si estamos seguros de querer borrar el elemento seleccionado, en caso afirmativo se hará una consulta a la base de datos, a ver si está asociado algún radiador, en este caso de estar en algún radiador mostraremos un mensaje e interrumpiremos el proceso de borrado.

En el caso de modificar, si el elemento seleccionado está asociado a algún radiador, mostraremos un mensaje en el que le informaremos de los radiadores afectados y si el usuario presiona sobre “Aceptar”, el proceso de modificación se realizará, con sus respectivos cambios en sus asignaciones.



Caso de Uso Gestión Diámetro de Entrada



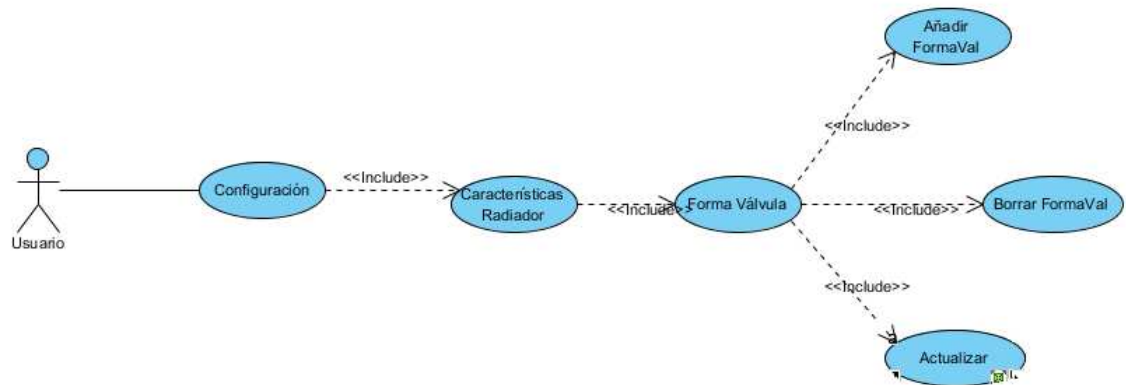
En este caso de uso podremos crear un diámetro de entrada del tubo, también podremos borrar el diámetro que seleccionemos y por último podremos modificar un diámetro, solo se nos permitirá cambiar la especificación del diámetro, nunca la clave primaria.

En el caso de borrar primero nos preguntará si estamos seguros de querer borrar el elemento seleccionado, en caso afirmativo se hará una consulta a la base de datos, a ver si está asociado algún radiador, en este caso de estar en algún radiador mostraremos un mensaje e interrumpiremos el proceso de borrado.

En el caso de modificar, si el elemento seleccionado está asociado a algún radiador, mostraremos un mensaje en el que le informaremos de los radiadores afectados y si el usuario presiona sobre “Aceptar”, el proceso de modificación se realizará, con sus respectivos cambios en sus asignaciones.



Caso Uso Gestión Forma Válvula



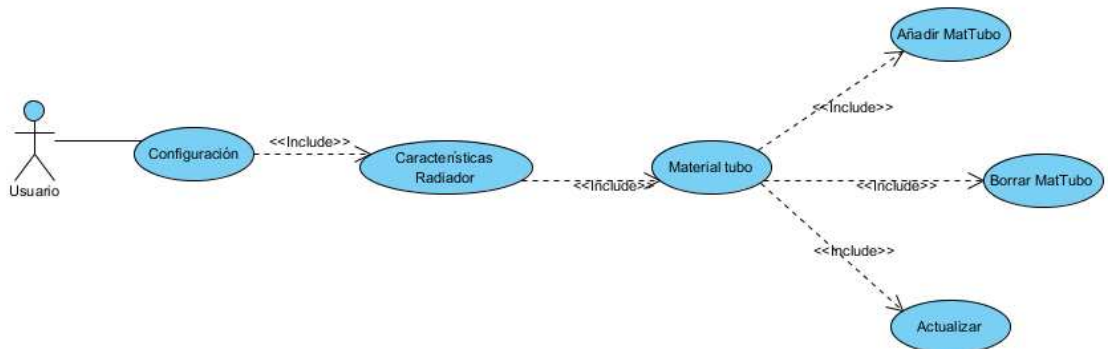
En este caso de uso podremos crear una forma de válvula, también podremos borrar la forma de la válvula que seleccionemos y por último podremos modificar una forma de válvula, solo se nos permitirá cambiar la especificación de la forma, nunca la clave primaria.

En el caso de borrar primero nos preguntará si estamos seguros de querer borrar el elemento seleccionado, en caso afirmativo se hará una consulta a la base de datos, a ver si está asociado algún radiador, en este caso de estar en algún radiador mostraremos un mensaje e interrumpiremos el proceso de borrado.

En el caso de modificar, si el elemento seleccionado está asociado a algún radiador, mostraremos un mensaje en el que le informaremos de los radiadores afectados y si el usuario presiona sobre “Aceptar”, el proceso de modificación se realizará, con sus respectivos cambios en sus asignaciones.



Caso de Uso Gestión Material Tubo



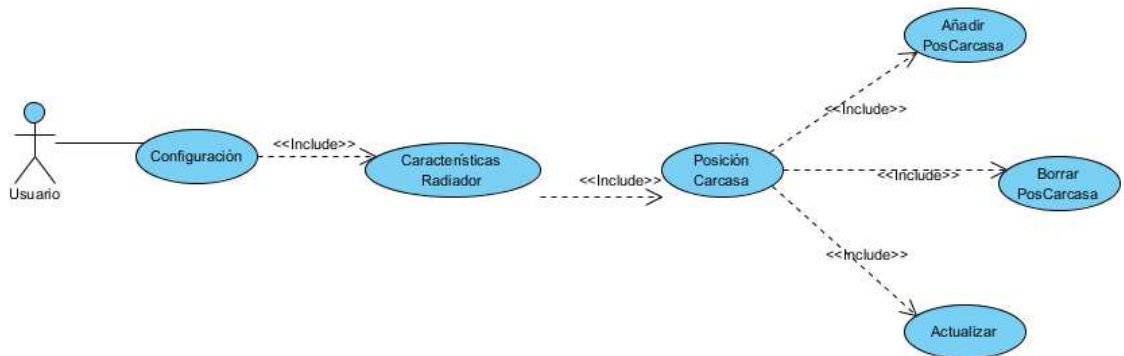
En este caso de uso podremos crear un material para el tubo, también podremos borrar el material que seleccionemos y por último podremos modificar los datos del material, solo se nos permitirá cambiar la especificación del material, nunca la clave primaria.

En el caso de borrar primero nos preguntará si estamos seguros de querer borrar el elemento seleccionado, en caso afirmativo se hará una consulta a la base de datos, a ver si está asociado algún radiador, en este caso de estar en algún radiador mostraremos un mensaje e interrumpiremos el proceso de borrado.

En el caso de modificar, si el elemento seleccionado está asociado a algún radiador, mostraremos un mensaje en el que le informaremos de los radiadores afectados y si el usuario presiona sobre “Aceptar”, el proceso de modificación se realizará, con sus respectivos cambios en sus asignaciones.



Caso de Uso Posición Carcasa



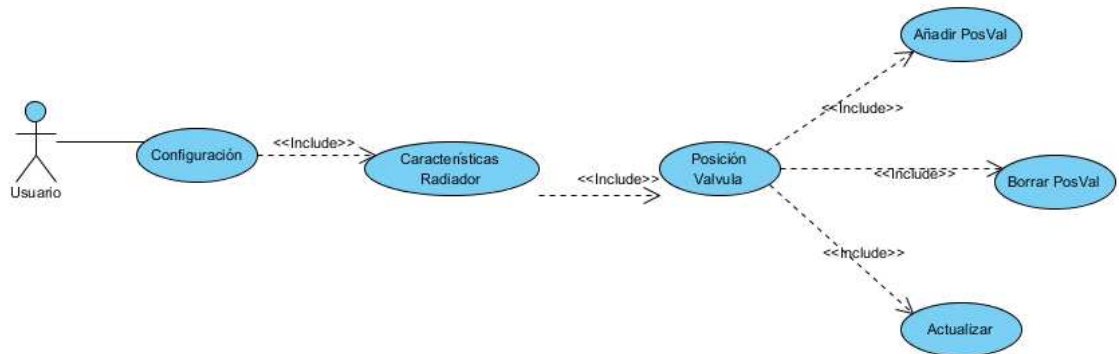
En este caso de uso podremos crear una posición para la carcasa, también podremos borrar la posición de la carcasa que seleccionemos y por último podremos modificar los datos de la posición de la carcasa, solo se nos permitirá cambiar la especificación de la posición, nunca la clave primaria.

En el caso de borrar primero nos preguntará si estamos seguros de querer borrar el elemento seleccionado, en caso afirmativo se hará una consulta a la base de datos, a ver si está asociado algún radiador, en este caso de estar en algún radiador mostraremos un mensaje e interrumpiremos el proceso de borrado.

En el caso de modificar, si el elemento seleccionado está asociado a algún radiador, mostraremos un mensaje en el que le informaremos de los radiadores afectados y si el usuario presiona sobre “Aceptar”, el proceso de modificación se realizará, con sus respectivos cambios en sus asignaciones.



Caso de Uso Gestión Posición Válvula



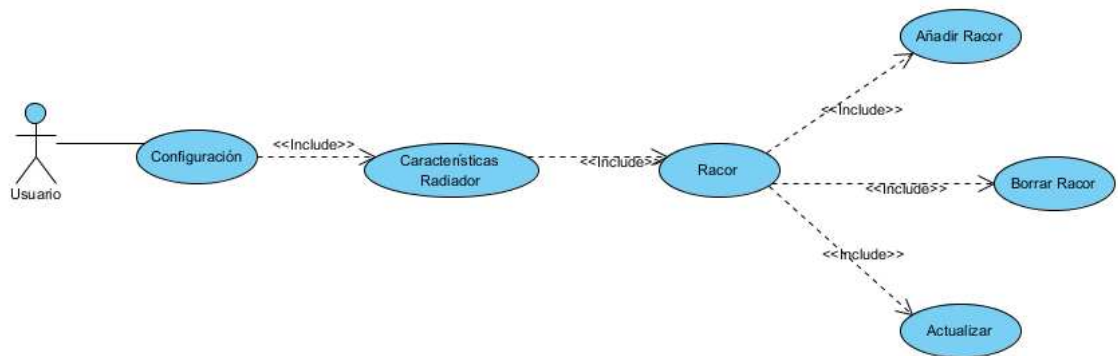
En este caso de uso podremos crear una posición para la válvula, también podremos borrar la posición de la válvula que seleccionemos y por último podremos modificar los datos de la posición de la válvula, solo se nos permitirá cambiar la especificación de la posición, nunca la clave primaria.

En el caso de borrar primero nos preguntará si estamos seguros de querer borrar el elemento seleccionado, en caso afirmativo se hará una consulta a la base de datos, a ver si está asociado algún radiador, en este caso de estar en algún radiador mostraremos un mensaje e interrumpiremos el proceso de borrado.

En el caso de modificar, si el elemento seleccionado está asociado a algún radiador, mostraremos un mensaje en el que le informaremos de los radiadores afectados y si el usuario presiona sobre “Aceptar”, el proceso de modificación se realizará, con sus respectivos cambios en sus asignaciones.



Caso de Uso Gestión Racor



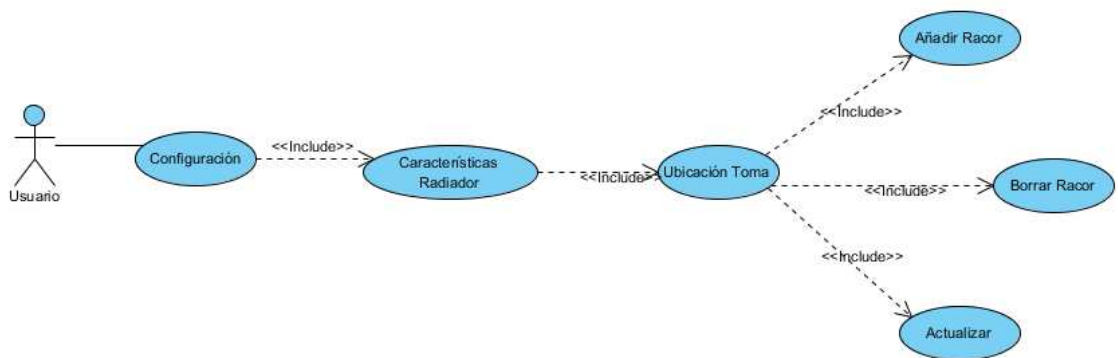
En este caso de uso podremos crear un racor nuevo, también podremos borrar el racor que seleccionemos y por último podremos modificar los datos del racor, solo se nos permitirá cambiar la especificación del racor, nunca la clave primaria.

En el caso de borrar primero nos preguntará si estamos seguros de querer borrar el elemento seleccionado, en caso afirmativo se hará una consulta a la base de datos, a ver si está asociado algún radiador, en este caso de estar en algún radiador mostraremos un mensaje e interrumpiremos el proceso de borrado.

En el caso de modificar, si el elemento seleccionado está asociado a algún radiador, mostraremos un mensaje en el que le informaremos de los radiadores afectados y si el usuario presiona sobre “Aceptar”, el proceso de modificación se realizará, con sus respectivos cambios en sus asignaciones.



Caso de Uso Gestión Ubicación Toma



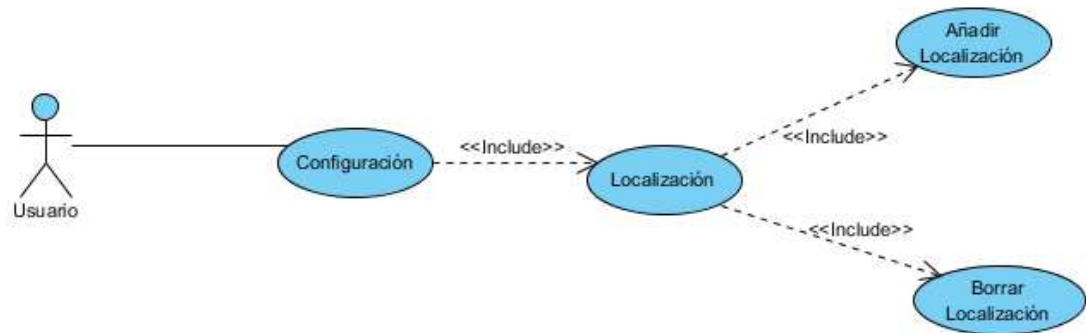
En este caso de uso podremos crear una ubicación para la toma del radiador, también podremos borrar la ubicación de la toma que seleccionemos y por último podremos modificar los datos de la ubicación de la toma, solo se nos permitirá cambiar la especificación del racor, nunca la clave primaria.

En el caso de borrar primero nos preguntará si estamos seguros de querer borrar el elemento seleccionado, en caso afirmativo se hará una consulta a la base de datos, a ver si está asociado algún radiador, en este caso de estar en algún radiador mostraremos un mensaje e interrumpiremos el proceso de borrado.

En el caso de modificar, si el elemento seleccionado está asociado a algún radiador, mostraremos un mensaje en el que le informaremos de los radiadores afectados y si el usuario presiona sobre “Aceptar”, el proceso de modificación se realizará, con sus respectivos cambios en sus asignaciones.



Caso de Uso oestión de Localización



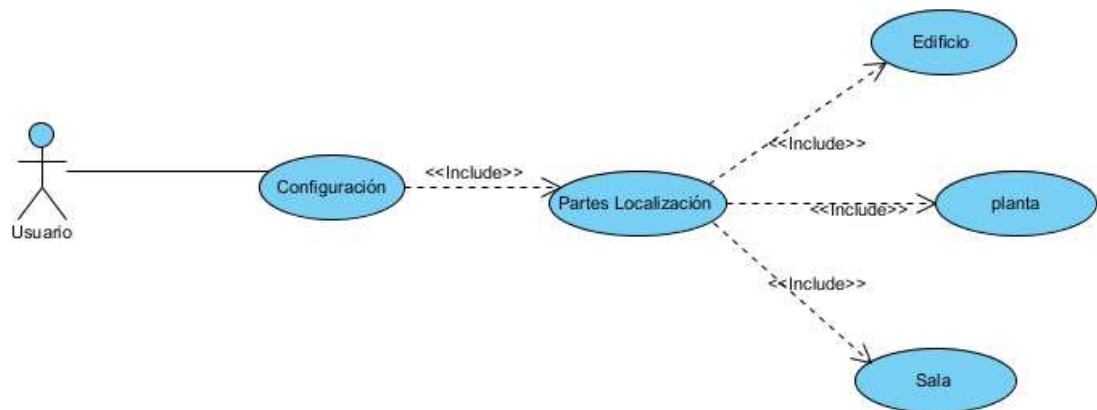
En este caso de uso podremos crear una localización para el radiador, también podremos borrar la ubicación de la toma que seleccionemos.

Una localización es la combinación de los valores de la tabla TEdificio, TPlanta y TSala, de esta manera ubicamos de forma física el radiador en una zona concreta del colegio.

En el caso de borrar primero nos preguntará si estamos seguros de querer borrar el elemento seleccionado, en caso afirmativo se hará una consulta a la base de datos, a ver si está asociado algún radiador, en este caso de estar en algún radiador mostraremos un mensaje e interrumpiremos el proceso de borrado.

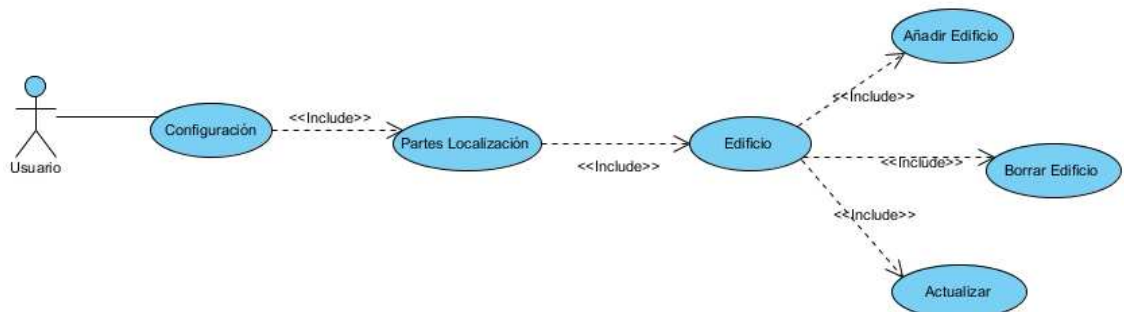


Caso de Uso Gestión de Partes Localización



En este caso de uso por medio de las diferentes pestañas podremos acceder a los datos de las tablas TEdificio, TPlanta y TSala.

Caso de Uso Gestión Edificio



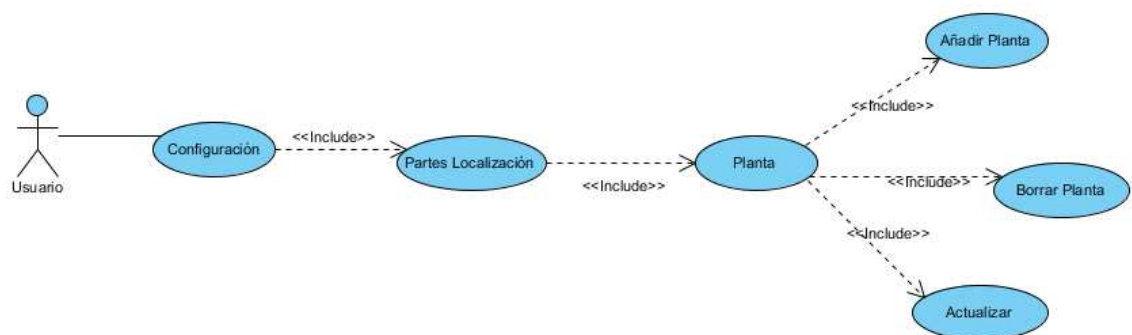


En este caso de uso podremos crear un edificio, también podremos borrar edificios y por último podremos modificar los datos de los edificios, solo se nos permitirá cambiar la especificación de dichos edificios, nunca la clave primaria.

En el caso de borrar primero nos preguntará si estamos seguros de querer borrar el elemento seleccionado, en caso afirmativo se hará una consulta a la base de datos, a ver si está asociado alguna localización, en este caso de estar en alguna localización mostraremos un mensaje e interrumpiremos el proceso de borrado.

En el caso de modificar, si el elemento seleccionado está asociado a alguna localización, mostraremos un mensaje en el que le informaremos de las localizaciones afectadas y si el usuario presiona sobre “Aceptar”, el proceso de modificación se realizará, con sus respectivos cambios en sus asignaciones.

Caso de Uso Gestión De Plantas



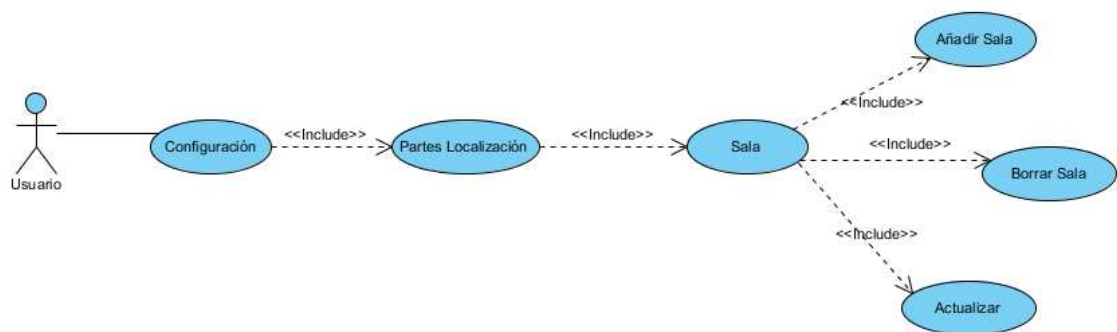
En este caso de uso podremos crear una planta, también podremos borrar plantas y por último podremos modificar los datos de las plantas, solo se nos permitirá cambiar la especificación de dichas plantas, nunca su clave primaria.

En el caso de borrar primero nos preguntará si estamos seguros de querer borrar el elemento seleccionado, en caso afirmativo se hará una consulta a la base de datos, a ver si está asociado alguna localización, en este caso de estar en alguna localización mostraremos un mensaje e interrumpiremos el proceso de borrado.



En el caso de modificar, si el elemento seleccionado está asociado a alguna localización, mostraremos un mensaje en el que le informaremos de las localizaciones afectadas y si el usuario presiona sobre “Aceptar”, el proceso de modificación se realizará, con sus respectivos cambios en sus asignaciones.

Caso de Uso Gestión Sala



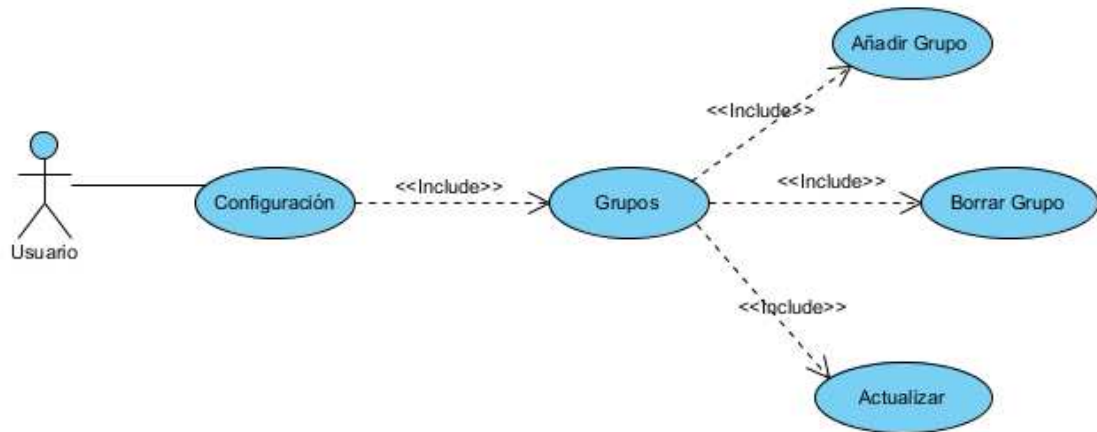
En este caso de uso podremos crear una sala, también podremos borrar salas y por último podremos modificar los datos de las salas, solo se nos permitirá cambiar la especificación de dichas salas, nunca la clave primaria.

En el caso de borrar primero nos preguntará si estamos seguros de querer borrar el elemento seleccionado, en caso afirmativo se hará una consulta a la base de datos, a ver si está asociado alguna localización, en este caso de estar en alguna localización mostraremos un mensaje e interrumpiremos el proceso de borrado.

En el caso de modificar, si el elemento seleccionado está asociado a alguna localización, mostraremos un mensaje en el que le informaremos de las localizaciones afectadas y si el usuario presiona sobre “Aceptar”, el proceso de modificación se realizará, con sus respectivos cambios en sus asignaciones.



Caso de Uso Gestión Grupos



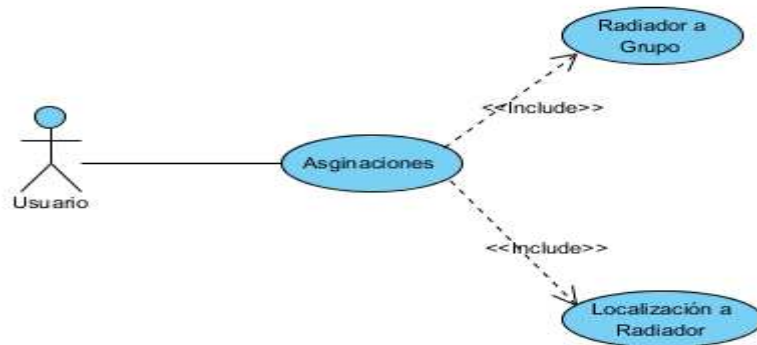
En este caso de uso podremos crear un grupo, también podremos borrar el grupo que seleccionemos y por último podremos modificar los datos del grupo, solo se nos permitirá cambiar la especificación del grupo, nunca la clave primaria.

En el caso de borrar primero nos preguntará si estamos seguros de querer borrar el elemento seleccionado, en caso afirmativo se hará una consulta a la base de datos, a ver si está asociado algún radiador, en este caso de estar en algún radiador mostraremos un mensaje e interrumpiremos el proceso de borrado.

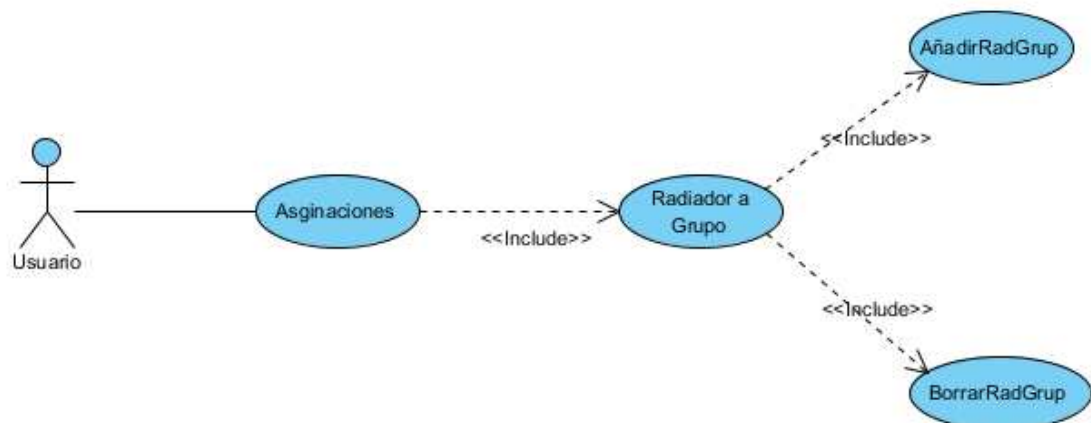
En el caso de modificar, si el elemento seleccionado está asociado a algún radiador, mostraremos un mensaje en el que le informaremos de los radiadores afectados y si el usuario presiona sobre “Aceptar”, el proceso de modificación se realizará, con sus respectivos cambios en sus asignaciones.



Caso de Uso Gestión de Asignaciones



Caso de Uso Gestión Radiador A Grupo

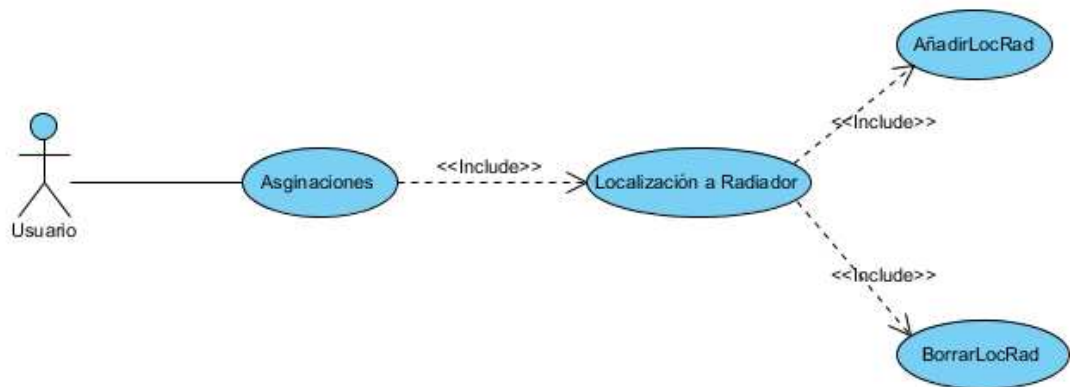


En este caso de uso podemos asignar radiadores a los grupo que hemos creado anteriormente.

Otra opción que se nos ofrece es la de sacar un radiador de un determinado grupo



Caso de Uso Gestión de Radiadores A Localización



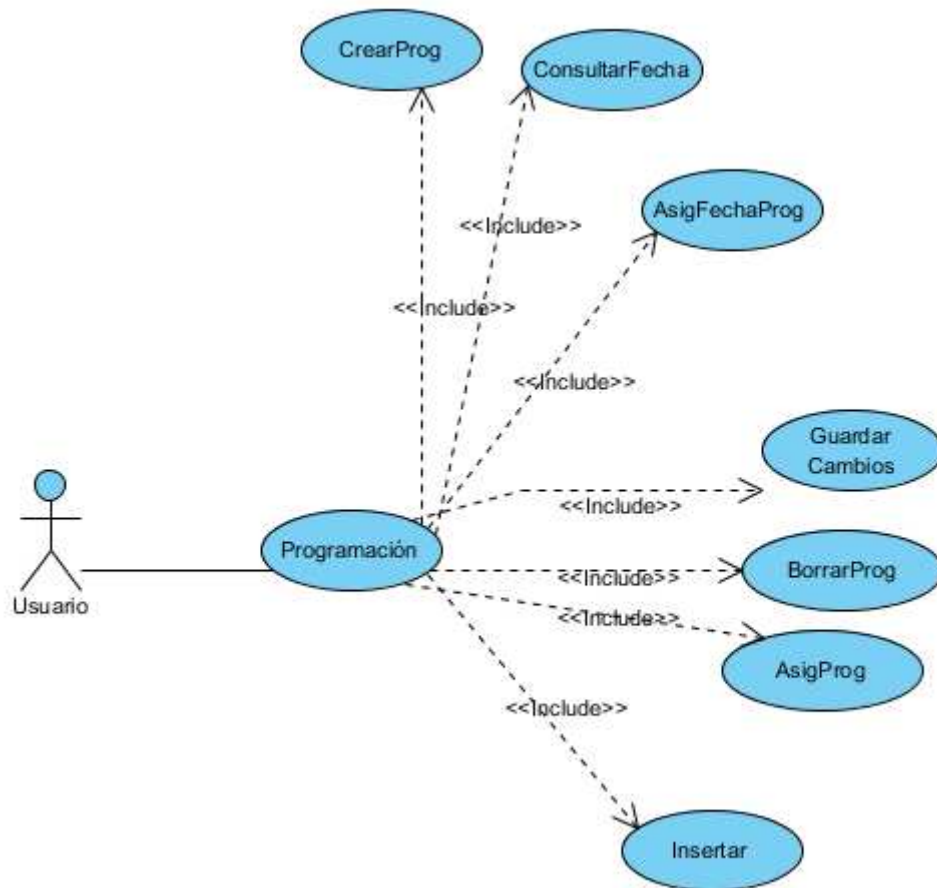
En esta pantalla es en la que nos encargamos de la ubicación física de nuestros radiadores, ya que le asignaremos el edificio, planta y sala (localización) en la que se encuentra.

Debido a que el concepto de localización es un poco abstracto para realizar esta inserción, mostraremos al usuario el la localización completa(edificio, planta y sala) en vez del identificador, de esta manera solo tendrá que seleccionar la que quiera y listo.

También se ofrecerá la posibilidad de desasignar a un radiador su localización



Caso de Uso Programación



Esta es una de las pantallas con más importancia de todo el proyecto ya que desde aquí se configuran todas las programaciones. Una programación va a tener un identificador, fecha de inicio y fin, los días de la semana en los que está activa y las temperaturas objetivo guardadas en intervalos de 30 min.

Una vez conocidas las características de una programación pasamos a explicar como configurar dichas programaciones. Las principales acciones a realizar son:

- Crear Programación

- Consultar intervalos de fechas para asignar a las programaciones



-Asignar el intervalo de duración de las fechas seleccionadas, además de poder modificar la fecha si se le agrega una fecha nueva, este proceso se podrá realizar siempre y cuando la programación no tenga ningún radiador asignado y en caso de tenerlo, que la nueva fecha no se solape con la fecha de alguna de las otras programaciones que tenga asignadas. En caso producirse solapamiento se mostrará un mensaje al usuario en el que se le indicará el radiador y la programación que está provocando el conflicto.

- Podremos modificar las temperaturas objetivo desde la tabla principal

-Borrar la programación que estemos visualizando (solo borraremos las temperaturas objetivo, guardaremos el identificador y el nombre de esa programación)

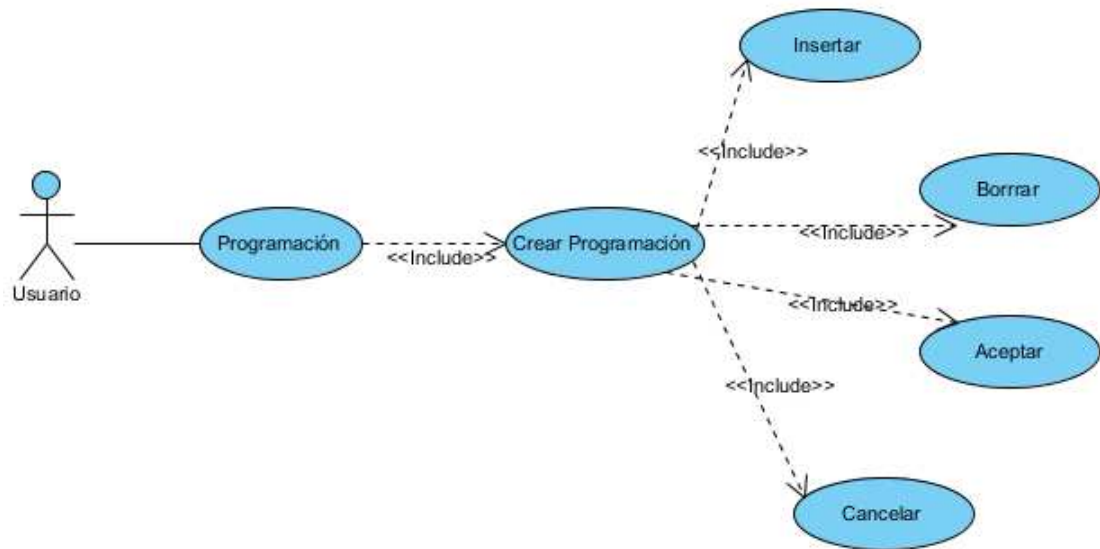
-Desde el botón AsigProg navegaremos a una ventana secundaria en la que tendremos la posibilidad de asignar todas las programaciones dadas de alta en nuestra BD a los radiadores. Un radiador puede tener varias programaciones siempre y cuando no se solapen las fechas. En caso de solapamiento se mostrará un mensaje de error.

-Desde insertar se introducirán las temperaturas objetivo que le hayamos indicado en el rango horario que hayamos marcado(en el proceso de diseño se detallará la información).

Ahora vamos a pasar a ahondar un poco mas en cada uno de los subcasos que nos ofrece esta pantalla.



Caso de Uso Gestión Crear Programación



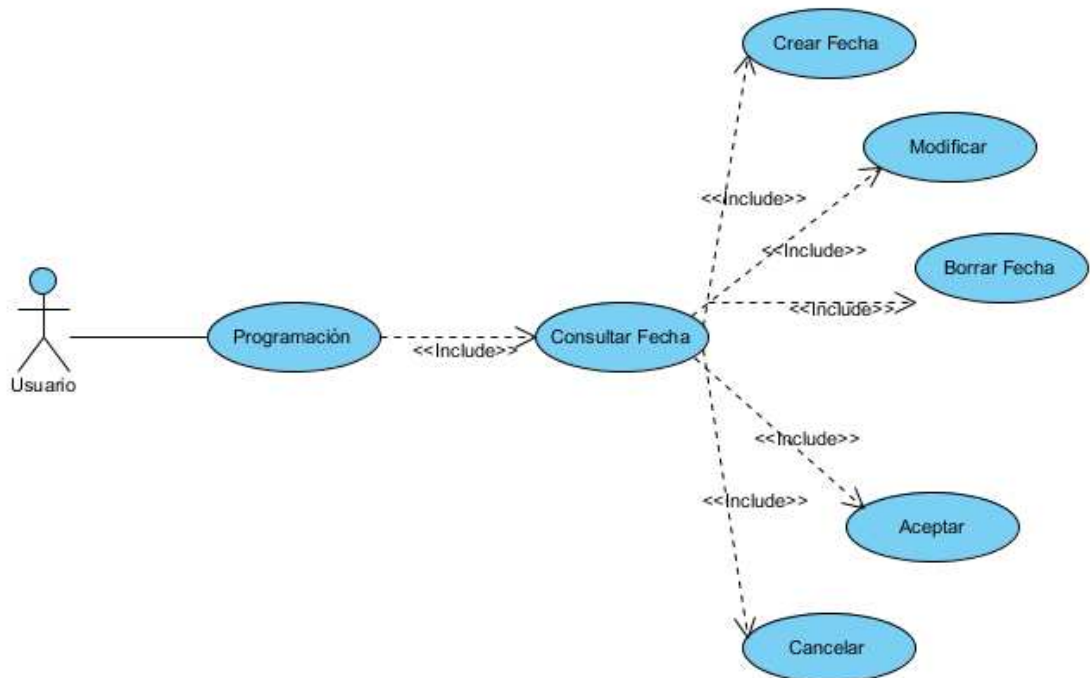
Este caso de uso aparece a partir de una ventana secundaria que surge al pinchar sobre el botón CrearProg. Aquí podrás darle nombre a una programación. Esto se hará dándole al botón insertar. Por convenio con la empresa todas las programaciones se crearán con todas las temperaturas objetivo a 15°.

Podremos borrar una programación pero no sin antes revisar que esta programación no tenga ningún radiador asignado, en caso de tenerlo se le mostrará avisará al usuario de este suceso y se le preguntara sobre la continuidad del proceso.

Tenemos 2 botones para salir de la ventana secundaria, si pinchamos sobre Aceptar saldremos de la ventana conservando la programación que hemos seleccionada (para futuras modificaciones) y si pinchamos en Cancelar saldremos sin conservar ningún valor.



Caso de Uso Gestión de Fechas

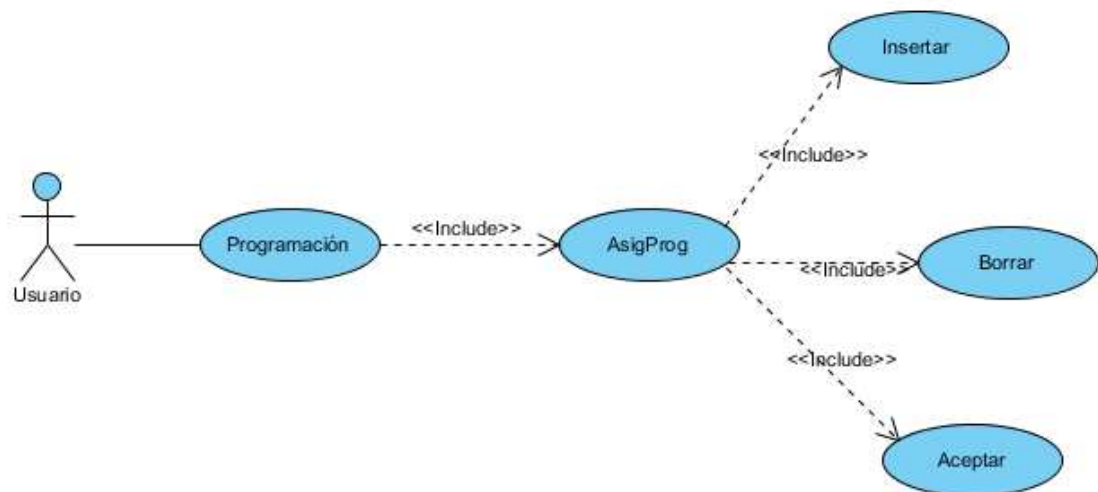


Aquí también navegaremos una ventana secundaria en la que podremos crear rangos de fechas, esto lo haremos seleccionando una fecha de inicio y una fecha de fin. Se podrá modificar fechas que ya estén creadas (esto será de gran utilidad ya que los calendarios de los colegios cambian de año en año, así nos evitamos crear toda la programación desde 0). Junto a crear y modificar tendremos la opción de borrar el intervalo de fechas que tengamos establecido.

Al igual que en la pantalla anterior, podremos salir de 2 formas diferentes, desde el botón de Aceptar o Cancelar. En este caso tiene bastante más sentido salir con aceptar puesto que nosotros trabajamos con el identificador de las fechas (un numero), ese numero a nosotros no nos suena de nada, pero si visualizamos la fecha de inicio y fin ya sabes con que fechas estamos trabajando de forma mas clara que un numero.



Caso de Uso Gestión de Asignación de Programaciones



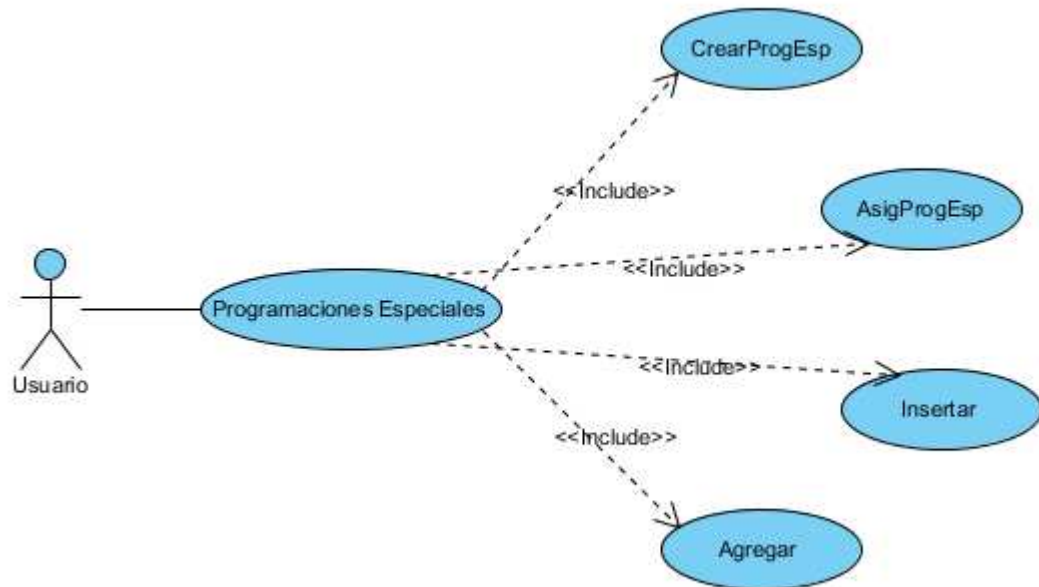
Aquí podremos asignar las programaciones creadas a los radiadores que seleccionemos. Aun no se ha hecho pero la inserción inicial de programaciones se hará a partir del Excel, dependiendo la localización y grupo al que pertenezca recibirá unas programaciones u otras. La asignación de programaciones se podrá realizar de forma individual o de forma colectiva, la segunda significa hacer las asignaciones por grupos (explicado en el caso de uso gestión de grupos), de esta forma todos los radiadores de características similares recibirán las mismas programaciones.

Ya que puede darse el caso en el que una programación no queremos que se siga ejecutando, tenemos la opción de borrar las asignaciones. Podemos hacer el borrado de 2 maneras, seleccionando la asignación en la tabla que muestra dichas asignaciones, o la otra forma es seleccionando un grupo y quitándole a grupo la programación seleccionada.

En este caso solo tenemos un botón para salir ya que una vez hayamos terminado de realizar las asignaciones no vamos a necesitar mas esos datos. El botón Aceptar nos devuelve a la página de programaciones



Caso De Uso Gestión de Programaciones Especiales



Debido a que en un colegio pueden darse imprevistos a lo largo del curso tenemos que tener la posibilidad de tener unas programaciones que tengan mas prioridad que otras. Si tenemos unas programaciones con prioridades evitamos tener que cambiar todas las programaciones y cuando se acabe esa situación especial volver a asignar todo. El tipo de situaciones especiales pueden ser del tipo clases extraescolares, grupo de alumnos que se marchan de excursión uno o más días, que en periodo vacaciones se abran las clases para actos públicos, días más fríos o más calidos...

Por todas estas situaciones nos vemos en la situación de crear unas programaciones especiales. Un radiador solo puede tener una única programación especial para evitar conflictos y tener que asignar niveles de prioridad.

Las programaciones especiales van a ser parecidas a las normales, la diferencia va a radicar en que las programaciones no van a tener días de la semana, se van a registrar por intervalos de fechas, esos intervalos van a tener una temperatura objetivo definida cada media hora.



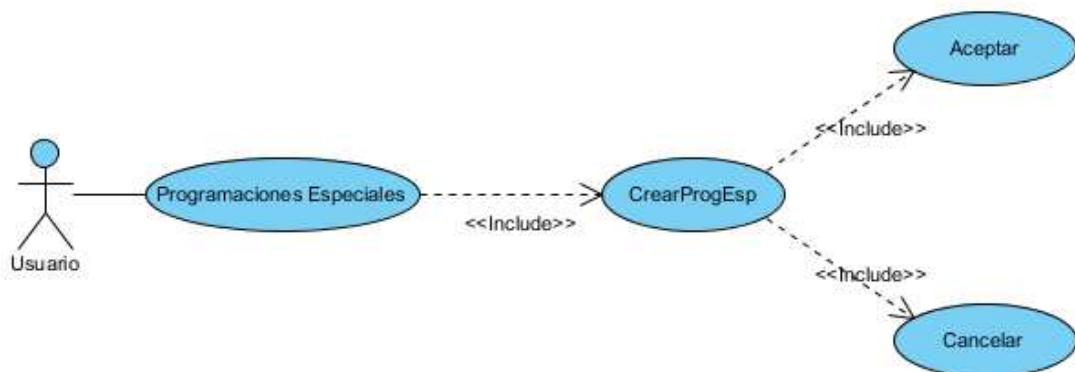
Desde esta pantalla presionando el botón de crear programación accederás a una ventana secundaria en la que tendremos la posibilidad de crear el intervalo de fechas que va a tener la programación especial. Una vez creamos las fechas se asignarán las temperaturas objetivo a 15° (por convenio con la empresa).

Desde el botón insertar podremos modificar las temperaturas objetivo, pondremos una hora de inicio y una hora de fin y a esa franja horaria definida le asignaremos la temperatura que nosotros deseemos.

Debido a que las programaciones especiales no tienen nombre, estas vienen con una breve descripción para saber por qué se crearon. Si seleccionamos una programación especial podremos visualizar su descripción, y si escribimos algo en el campo de texto en el que la estamos visualizando y pinchamos en agregar, la definición se verá modificada.

Por ultimo una vez tenemos la programación especial totalmente definida (fecha de inicio y fin, temperaturas objetivo definidas y la definición a nuestro gusto) ya podemos pasar a asignársela a algún radiador. Esto se hará presionando sobre el botón AsigProgEsp, este nos llevará a una ventana secundaria en la que podremos realizar la asignación.

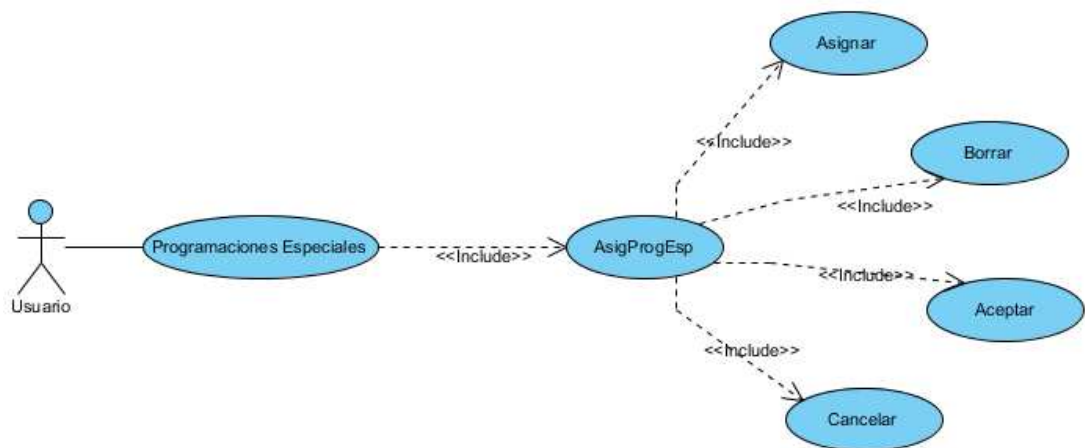
Caso de Uso Gestión de Creación Programación Especial





Este caso de uso es muy simple, solo tiene un botón de Aceptar o cancelar, si presionamos aceptar se creará la programación especial con la fecha que hayamos definido en esa ventana y con temperaturas objetivo a 15°, pero si salimos presionando sobre cancelar no haremos nada.

Caso de Uso Gestión de Asignación Programación Especial



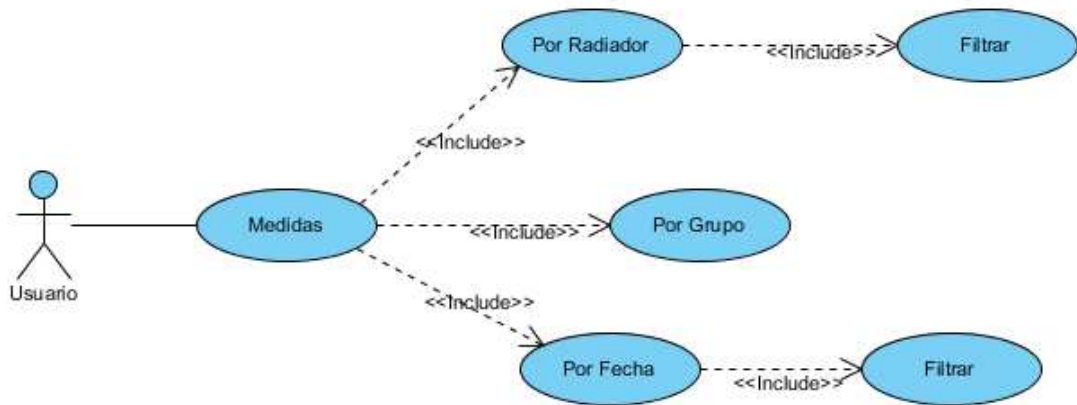
Desde aquí podremos visualizar que programaciones especiales tiene cada radiador, recordar que un radiador solo tiene una programación especial. Seleccionaremos el radiador y la programación a asignar y con el botón asignar realizaremos el proceso de asignación. En caso de ya tener una programación definida se preguntará al usuario si desea modificar la programación e insertar la nueva programación.

Podremos quitar la asignación de una programación especial si presionamos sobre el botón de borrar, siempre y cuando tengamos una programación seleccionada.

En esta ventana podemos salir tanto con el botón Aceptar como con el de Cancelar (se aplicará alguna diferencia en líneas futuras del proyecto)



Caso de Uso Gestión de Medidas



Desde esta pantalla podremos ver todas las medidas reflejadas por los sensores, como la cantidad de datos que vamos a tener va a ser muy grande (debido a que la media de radiadores de un colegio es de 200 y estos envían datos cada 5 min.) hemos buscado la forma de poder visualizarlos de forma mas fácil.

Hemos creado 3 tipos de agrupaciones:

-Por Radiador: Este agrupamiento nos muestra todas las medidas reflejadas por cada radiador. Además de mostrárnoslo agrupado radiadores nos da la opción de aplicar un filtro. Le podemos decir de que a que radiador queremos visualizar los datos. Si no introducimos valor mínimo suponemos que es 0 o si no ponemos valor máximo suponemos que queremos visualizar hasta el último. Muy útil para la detección de errores del tipo ver en que momento ha dejado de funcionar ese sensor, ya que tendremos todo su histórico ordenado por fecha.

-Por Grupo: Aquí nos mostrará los datos agrupados por grupo (si un radiador aparece en mas de un grupo, los datos de este radiador aparecerán mas de 1 vez) esto es útil para detectar error ya que si han reflejado temperaturas similares en los radiadores del mismo grupo todo va bien, pero si vemos que radiadores del mismo grupo tienen temperaturas muy diferentes, algo está ocurriendo, ya que se supone que están en situaciones cercanas y deberían trabajar igual.



-Por Fecha: Aquí nos agrupará todas las mediadas por fecha, el comportamiento de este filtro es similar al de por radiador. Será interesante para ver el comportamiento de los radiadores en intervalos de tiempo, por ejemplo si una semana ha hecho mucho frío ver como han trabajado esos radiadores durante esa semana.



4.5.3 Diseño Protocolo Comunicación

Protocolo Win USB:

Para realizar la comunicación entre nuestro programa y el comunicador hemos optado por transferir los datos por USB, hemos utilizado el puerto USB porque a día de hoy es el puerto más estandarizado y más fácil de encortar en todos los ordenadores.

Para esta comunicación se va a utilizar el driver WINUSB, es un driver genérico para Windows diseñado para la comunicación entre un Device y un Host.

El dispositivo Host siempre debe de empezar la comunicación entre los 2 dispositivos.

Aplicación a realizar:

- Device (coordinador de la red ZigBee) manda datos sobre las características de un radiador en concreto al Host (PC)

- Host contesta al coordinador con otros datos específicos del mismo radiador.

Como la comunicación tiene que ser inicializada por el Host(PC), será necesario que este dispositivo mande periódicamente un mensaje hasta que reciba respuesta del device(coordinador de la red).

Tipo de transferencia de datos

Transferencia de control(message pipe):

- Usada para configurar nuevos dispositivos en el bus
- Contenido de los paquetes exige: Petición/Respuesta/Estado
- No hay restricciones ni de frecuencia ni de latencia.
- Exige control de errores.



Transferencia Bulk(stream pipe):

- Grandes cantidades de datos de forma esporádica
- Solo se garantiza el correcto envío y recepción de datos.
- No garantiza límite de latencia ni de ancho de banda.

Transferencia Interrupción(stream pipe):

- Transferencia de pequeñas longitudes y esporádicas.
- Exige respuesta inmediata, lo que impone restricciones de latencia.
- Exige control de errores, retransmisión si es necesario.

Transferencia asíncrona:

- Información sin formato
- Generada y consumida en gran cantidad y de forma continua.
- Sólo se negocia el tamaño máximo de paquete.
- No se exige control de errores.
- Exige restricción de latencia(temporización es importante).
- USB garantiza una tasa constante de transferencia durante toda la transacción.

En nuestro caso se las transmisiones más adecuadas será las tipo bulk o tipo interrupción, probaremos con las bulk a ver qué tal funcionan las comunicaciones.

Para realizar la conexión entre el ordenador y la placa hemos utilizado el protocolo de comunicación WinUSB, este protocolo de comunicación consiste en 2 simples pasos, primero el Host (ordenador) inicia la comunicación con el device (placa), inmediatamente después de recibir este mensaje el device envía otro mensaje al Host.

Nosotros hemos tenido que realizar algunos cambios en este protocolo debido a que nosotros necesitábamos que el device tenía que iniciar la comunicación y el Host tenía que contestar, justo al revés que el protocolo WinUSB. Para solucionar este



problema nosotros hemos creado un protocolo de comunicación además de usar el WinUSB. Debido que siempre el Host va tener que inicializar la conversación hemos dividido dicha conversación en 5 partes.

1º) El Host enviará un mensaje de control al device, este mensaje consistirá en un mensaje en el que el primer byte será 0x10, este mensaje significará que queremos iniciar la conversación.

2º) El device al recibir el mensaje de control y haber recibido información de alguno de los radiadores de la red, solo si se cumplen las 2 condiciones responderá con otro mensaje en el que el primer byte será 0x20, los demás bytes serán la información que se este registrando en el sensor.

La estructura del mensaje enviado del device al Host será:

Byte0 : Tipo de mensaje, lo vamos a definir como 0x20

Byte1-2: Radiador: irá del número 0 al 65535

Byte1-menos significativo

Byte2- más significativo

Byte 3-4: Temperatura entrada:

Resolución de 10 bits: 0bits es 0º y 1024bits es 100º

Byte3-menos significativo

Byte4- más significativo

Byte 5-6: Temperatura de salida:

Resolución de 10 bits: 0bits es 0º y 1024 bits es 100º

Byte5-menos significativo

Byte6- más significativo

Byte 7-8: Temperatura entorno radiador



Resolución de 10 bits: 0bits es 0° y 1024bits es 100°

Byte7- menos significativo

Byte8- más significativo

Byte 9-10: Temperatura ambiente

Resolución de 10 bits: 0 bits es 0° y 1024 es 100°

Byte9- menos significativo

Byte10- más significativo

Byte 11-12: Voltaje del supercondensador

Resolución de 10 bits: 0 bits es 0V y 1024 bits es 3V

Byte11- menos significativo

Byte12- más significativo

3º) El Host al recibir el mensaje del device con el primer byte 0x20, se encargará de enviar a la placa un mensaje en el que el primer byte esté a 0x30, en los otros bytes enviaremos la temperatura a la que tiene que estar el radiador con el cual estamos teniendo la conversación.

La estructura del mensaje enviado del Host al device será:

Byte0 : Tipo de mensaje, lo vamos a definir como 0x30

Byte1-2: Radiador: irá del número 0 al 65535

Byte1- menos significativo

Byte2- más significativo

Byte 3-4: Temperatura objetivo:

Resolución de 10 bits: 0bits es 0° y 1024bits es 100°

Byte3- menos significativo

Byte4- más significativo



4º) La placa una vez recibe el mensaje con el 0x30, responderá a ordenador con otro en el que su primer byte será 0x40, esto significará que todo el proceso de envío a sido correcto y ya puede volver a enviar un nuevo mensaje de control para iniciar una nueva conversación .

5º) Como los dispositivos están dormidos la mayoría del tiempo, puede que enviemos el mensaje de control y no tengamos datos a leer, cuando el Host reciba un mensaje en el que el primer byte está a 0x50, significa que no hay datos para enviar por parte de la placa. Entonces tendremos que volver a reiniciar la conversación volviendo a enviar el mensaje de control.

El funcionamiento del programa consistirá en enviar mensajes de control (byte0=0x10) y ver lo que nos ha respondido el comunicador, si el comunicador no está listo, ya sea por no tener datos que enviar o porque está atendiendo la petición de algún sensor, este nos responderá con un 0x50 en el byte0, si ocurre esto volveremos a enviar otro mensaje de control, pero si por el contrario este tiene datos que enviarnos nos responderá con un 0x20 en el byte 0.

En el momento que recibimos el 0x20 insertamos en la base de datos los datos recibidos del comunicador. Después de insertar los datos correspondientes pasaremos a ver que programación tiene el sensor que nos ha enviado los datos.

Una vez conocida la programación (en caso de no tener programación asignada el sensor funciona de forma autónoma y trabajará con la temperatura que se le asignase cuando se programó) enviaremos un mensaje con el byte0 a 0x30 y la temperatura objetivo que queremos que trabaje el radiador con el que estamos manteniendo la comunicación.

Tras haber enviado el mensaje con la temperatura objetivo nos quedamos esperando a que el comunicación nos conteste con un 0x40 en el byte0. Después de recibir el 0x40 volvemos a iniciar el proceso de comunicación, es decir, volvemos a enviar un mensaje de control (byte0=0x10).



El pseudocódigo quedaría así:

```
while ( true)
{
    databufferEnv[0] = 0x10;
    enviar_USB(databufferEnv);
    leido_20 = false;
    leido_40 = false;
    databufferRead= leer_USB();
    if (databufferRead [0] == 0x50)
    {
        enviar_USB(databufferEnv);
    }
    else
    {
        while (leido_20 == false)
        {
            if(databufferRead == 0x20)
            {
                leido_20=true;
            }
            else
            {
                databufferRead= leer_USB();
            }
        }

        Insertar_BD(databufferRead);
        temp=Consultar_Prog_BD(databufferRead);
        databufferEnv[0]=0x30;
        databufferEnv[2]=Temp.;
        enviar_USB(databufferEnv);

        while (leido_40 == false)
        {
            if(databufferRead == 0x40)
            {
                leido_40=true;
            }
            else
```



```
        {  
            databufferRead= leer_USB();  
        }  
    }  
  
}  
  
}
```

Cuando hacemos el segundo envío no miramos si el comunicador nos responde con un 0x50 porque una vez hemos hecho ya un envío el comunicador se queda atendiendo nuestras peticiones hasta el final de la comunicación(nos envíe el 0x40)

Principales funciones del Programa:

Insertar_BD(Byte[] buffer)

En esta función vamos a recibir como parámetro el buffer que hayamos leído del comunicador. Lo primero que haremos será la conversión de los bytes del buffer a decimal para ver que radiador es el que nos está enviando la información.

```
int idRad;  
idRad = bufferReciv[1] + bufferReciv[2] * 256;
```

Después de saber cual es el radiador del que estamos recibiendo la información, descodificaremos los datos recibidos del comunicador(temperatura ambiente, entorno, entrada, salida, voltaje SuperCondensador, peltier y posición motor, estas dos ultimas medidas no se envían de momento, pero en un futuro no muy lejano se registrarán estos datos para estudios de consumo y optimización) ya que nos llegan en forma de bytes en hexadecimal y nosotros trabajamos con decimales. Un ejemplo de codificación de temperatura es:

```
double tempEnt;  
string tempEntInsetar;
```

```
tempEnt = bufferReciv[3] + bufferReciv[4] * 256;  
tempEntInsetar = ((tempEnt * 100) / 1024).ToString("N2");
```



Las otras 3 medidas de temperaturas siguen el mismo proceso. El N2 significa coger los 2 primeros decimales (el porque de las multiplicaciones y divisiones explicado arriba, en la parte que se explica el valor de cada byte).

Ya que los voltajes tienen una conversión diferente a las temperaturas vamos a poner un ejemplo:

```
double volSC;  
string volSCInsertar;  
  
volSC = bufferReciv[11] + bufferReciv[12] * 256;  
volSCInsertar = ((volSC * 3) / 1024).ToString("N2");
```

Una vez que ya tenemos todos los datos listos para insertar hacemos una consulta para ver cuál ha sido la última medida registrada en la base datos, para insertarlo en el siguiente. Las consultas en esta parte del proyecto son en C#. En cuanto tenemos el id de la ultima medida ya estamos en disposición de realizar la inserción en la base de datos.

consultaProg_BD(Byte[] buffer)

Esta función la podemos dividir en 2 partes, primero realizaremos una consulta en la base de datos a ver si el radiador con el que estamos trabajando tiene asignada una programación especial o no. En caso de tener una programación especial asignada me quedo con todas aquellas cuyo periodo de acción coincide con el día de hoy. Si después de estas 2 condiciones la consulta nos devuelve algo es que tenemos que ejecutar una programación especial, ahora solo nos queda ver en que hora del día nos encontramos para enviarle a comunicador la temperatura objetivo.

Pero si la consulta anterior nos devuelve 0 es porque o no tenemos una programación especial asignada o porque la el intervalo de actuación de esa fecha no es el actual. Tras descartar la posibilidad de tener programaciones especiales nos



centramos en las programaciones semanales. Lo primero que haremos será mirar en que día de la semana nos encontramos.

Vamos a hacer la consulta en 3 partes, para ir filtrando los datos y quedarnos justo con el que queremos. En un principio vamos a sacar todas las programaciones que tiene el radiador con el que estamos manteniendo la conversación. Para saber el valor del radiador es el mismo proceso que hemos seguido en la función `insertar_BD()`.

Una vez tenemos todas las programaciones del radiador en cuestión .Ahora nos quedaremos con todos los identificadores de las fechas en los cuales el día de inicio sea menor que el día de hoy y la fecha de fin sea mayor a la de hoy(es decir, este dentro de su rango de acción).

Una vez tenemos los datos del día en el que estamos, el id de la fecha calculado anteriormente y el id de la programación, ya podemos obtener la temperatura. La temperatura la tendremos mirando la hora del día y acotando las horas cada 30 min.

Después de esto ya tenemos la temperatura objetivo lista para enviar al device.

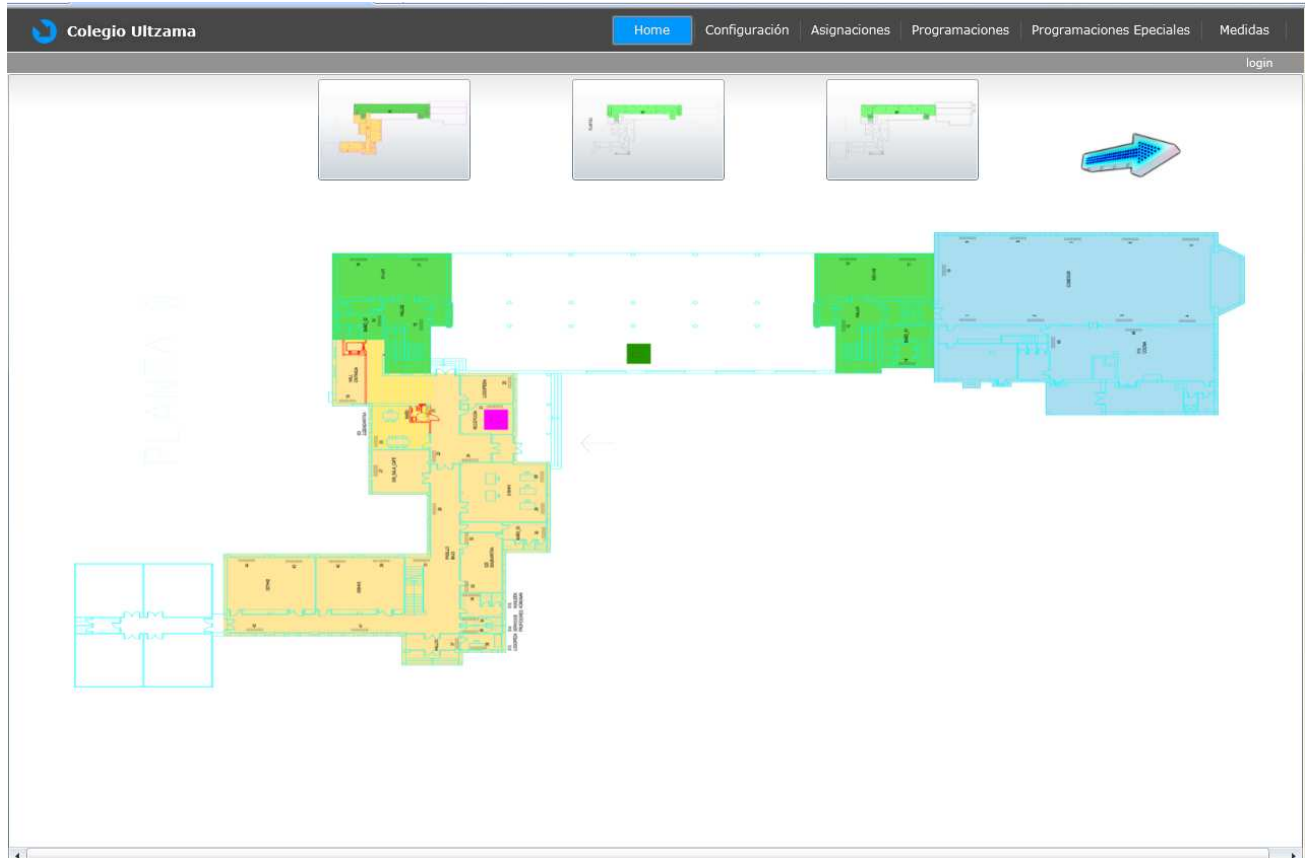


4.5.4 Implementación

Una vez que hemos terminado todo el periodo de análisis y diseño pasaremos a implementar. Lo primero que se hizo fue crear en papel un borrador de las principales pantallas de nuestro proyecto, tras crear estos primeros borradores se realizó una reunión con la empresa interesada en el proyecto para recibir el visto bueno de la distribución de las cosas.

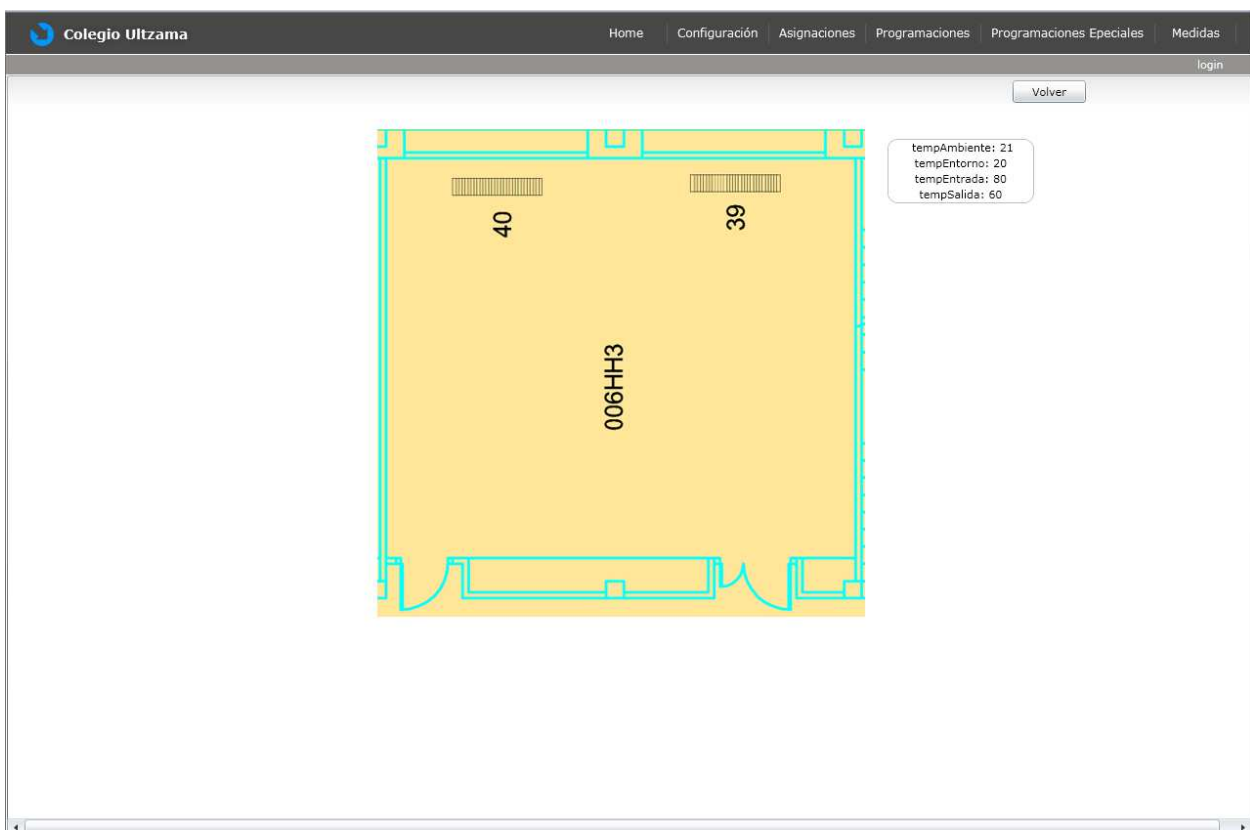
Una vez tuvimos el permiso de la empresa se comenzó a implementar todo. Del borrador inicial a resultado final hubo bastantes cambios ya que cada vez se intentaba refinar mas cosa para conseguir la máxima facilidad de uso. Al final las pantallas quedaron muy intuitivas y amigables, esta es una de las características que se buscaban desde un principio puesto que el usuario.

A continuación mostraremos algunas de las principales pantallas de nuestro proyecto





Esta en nuestra página de inicio, desde ella podemos navegar a todas las demás. En ella podemos navegar a todas y cada una de las plantas de nuestro colegio, podremos hacerlo por medio de la flecha de desplazamiento o por medio de los botones que representan las demás plantas. También se ofrece la posibilidad de hacer zoom sobre todas y cada una de las habitaciones del colegio, la finalidad del zoom es poder ver donde se encuentran ubicados los radiadores del colegio.



Una vez hacemos click en una habitación deseada obtendremos información acerca de los radiadores de esa habitación. Si pasamos con el cursor sobre el símbolo del radiador(rectángulo rallado) se mostrará un popup(elemento emergente) en el que se mostrará la ultima temperatura reflejada por ese radiador.

Además si pinchamos sobre ese radiador nos mostrará las programaciones que tiene asignadas.



Colegio Ultzama

HomeConfiguraciónAsignacionesProgramacionesProgramaciones EspecialesMedidas

login

RadiadorCaracterísticas RadiadorLocalizaciónPartes LocalizaciónGrupos

Radiador	Angulo Val	Diametro Ent	Forma Val	Material Tub	Pos Carcasa	Pos Val	Racor	Ubicacion Toma	Observaciones	Referencia	Numero Serie
2	180	34	E	PM	B	I	34	P	ninguna	fasda	1
3	180	34	E	PM	A	D	34	P	ninguna	fasda	2
15	270	34	E	CM	A	D	12	P	esto es una buer	prueba	3
16	270	34	E	PM	A	I	12	P	prueba 1 seg	bueno	4
39	180	34	E	CM	A	I	12	P	Primer radiador	NDFSWEFS	5
40	270	34	E	CM	B	D	34	P	asas	SSADASD	6
43	180	34	E	CM	A	D	34	P	Hola k tal	KJMNJI	7
44	180	34	E	CM	A	D	12	P		ESSFCE	8
46	270	34	E	CM	A	D	12	P			9
3	254	0.5	E	H	A	D	0.5	S		ASED-0.5-254H-0.5	123456789012
4	254	0.5	E	H	A	D	0.5	S		ASED-0.5-254H-0.5	123456789013

AñadirRadiador

DarBajaRadiador

BorrarRadiador

Actualizar

AbrirExcel

InsertarExcel

Esta es la página de configuración de radiadores, aquí podemos crear, borrar, modificar y dar de baja un radiador(explicado la forma de uso en los casos de uso). También podemos abrir excels e insertar los Excel abiertos.



Colegio Ultzama Home Configuración Asignaciones Programaciones Programaciones Especiales Medidas login

Radiador Características Radiador Localización Partes Localización Grupos

Angulo Valvula Diametro Entrada Forma Valvula Material Tubo Posicion Carcasa Posicion Valvula Racor Ubicacion Toma

Angulo Val	Categoria
180	normal
254	especial
270	normal

AñadirAngulo

BorrarAngulo

Actualizar

Radiador Características Radiador Localización Partes Localización Grupos

Angulo Valvula Diametro Entrada Forma Valvula Material Tubo Posicion Carcasa Posicion Valvula Racor Ubicacion Toma

Angulo Val	Categoria
180	normal
254	especial
270	normal

AñadirAngulo idAngulo

BorrarAngulo Categoria

Actualizar Insertar

Desde aquí podemos modificar todas las características de los radiadores. Si pinchamos sobre añadir ángulo (mismo mecanismos para todos los demás) se mostrará 2 campos de texto para introducir el ángulo y la clave de ese ángulo. Para que esta instrucción tenga efecto presionamos sobre el botón insertar



Colegio Ultzama Home Configuración Asignaciones **Programaciones** Programaciones Especiales Medidas login

Programacion

Fecha Nombre de la Programación: ProgInvierno FechaInicio: 02/06/2011 FechaFin: 30/06/2011

	Id Fecha	Id Prog	00:00 ->	00:30 ->	01:00 ->	01:30 ->	02:00 ->	02:30 ->	03:00 ->	03:30 ->	04:00 ->	04:30 ->	05:00 ->	05:30 ->	06:00 ->	06:30 ->	07
Lunes	1	65	12	12	25	12	12	12	15	15	15	15	15	23	23	23	23
Martes	1	65	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15
Miércoles	1	65	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15
Jueves	1	65	15	15	15	15	8	9	10	15	15	15	15	15	15	15	15
Viernes	1	65	15	15	12	11	3	2	4	5	15	15	15	15	15	15	15
Sábado	1	65	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15
Domingo	1	65	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15

Lunes

De:

a

o

Martes

De:

a

o

Miércoles

De:

a

o

Jueves

De:

a

o

Viernes

De:

a

o

Sábado

De:

a

o

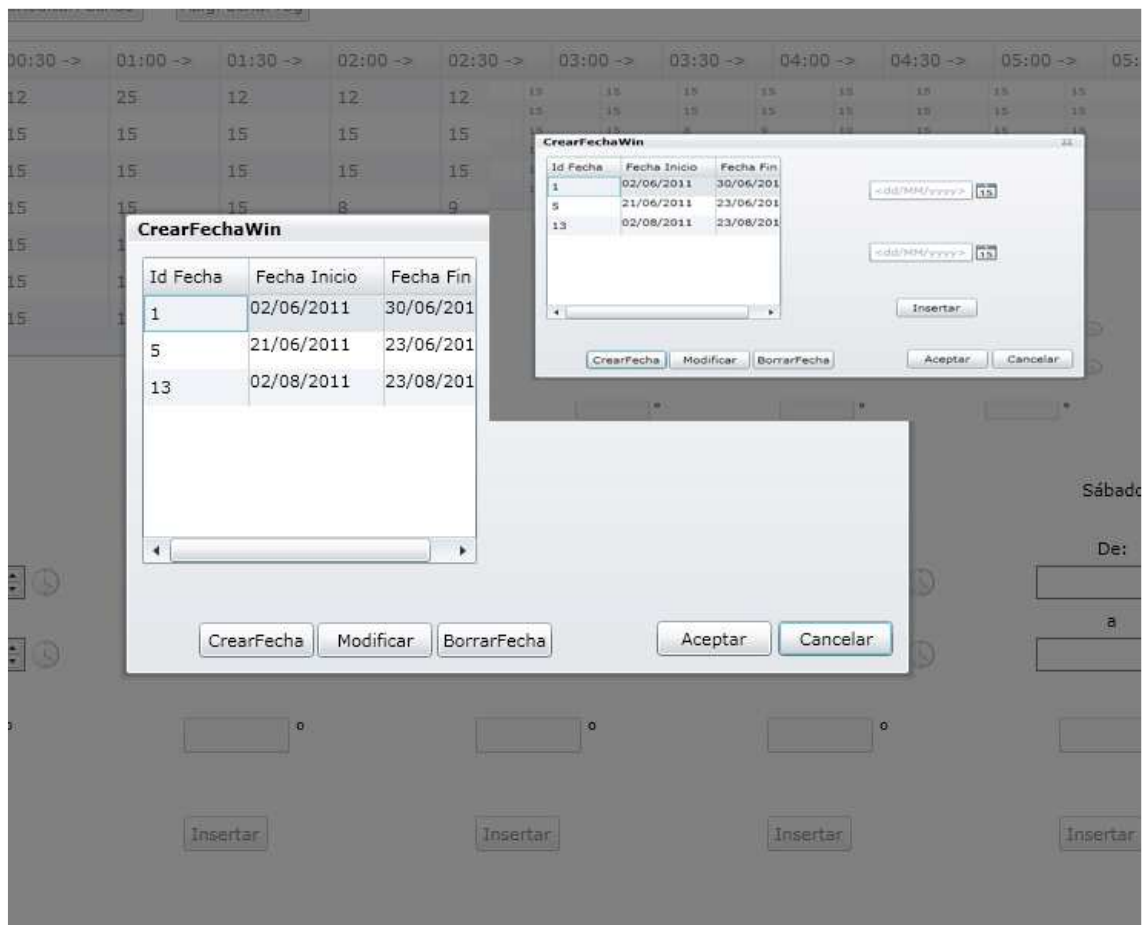
Domingo

De:

a

o

Esta es la página mas importante del proyecto ya que en ella se definen las programaciones que van a estar funcionando a lo largo del calendario escolar. Seleccionaremos la programación a visualizar desde el combobox(desplegable junto a la palabra programación). Desde los relojes podemos definir intervalos de tiempo y según la temperatura que pongamos debajo se insertará en nuestra programación. Todo este proceso se realizará tras presionar sobre el botón de insertar. El mecanismo es el mismo para todos los días. Si presionamos sobre CrearProg, ConsultarFecha y AsignarProg se abrirá una venta emergente de la que obtendremos diferentes tipos de información)



Este es un ejemplo de las ventanas que salen de presionar alguno de los botones mencionados anteriormente. Todas tienen un comportamiento similar. Para crear una nueva fecha tenemos que hacer click sobre crearFecha y aparecerán dos datapiker (calendarios) en los que seleccionar el rango del nuevo intervalo de fechas.



4.6 Pruebas

Este es uno de los pasos mas complicados del proyecto ya que tenemos que hacer interactuar nuestro proyecto con el otro. Es más nuestro proyecto también consta de 2 módulos diferenciados, los cuales tendremos interactuar.

Comunicación entre nuestros módulos:

-Verificar la conexión con la base de datos entre el módulo principal que es el que tiene la base de datos y el modulo secundario.

- Verificar el correcto acceso a los datos de la BD

-Verificar correcta inserción de los datos

Comunicación con el otro proyecto:

-Verificar la conexión con el device

-Verificar el correcto uso del protocolo de comunicación que hemos desarrollado

- Correcto envío del mensaje de control

- Correcta recepción de los mensajes del device

- Correcto envío de temperaturas objetivo

- Recepción de mensaje de fin de comunicación

Tenemos que comprobar que todos esto pasos se cumplen como nosotros le hemos dicho en diferentes tipos de situaciones

-Trataremos de probar los casos más remotos y poco habituales porque siempre se puede dar el caso de que exista y por lo tanto lo tenemos que tener bajo control.

-Verificar que cada radiador reflejas sus medidas y no las de otro.

-Verificar correcto funcionamiento de las programaciones asignadas(en cada momento envía la temperatura que nosotros le hemos asignado)



Pruebas de Unidad

Este tipo de pruebas consiste en mirar que al cambiar algo en la base de datos no se produzcan inconsistencia en los datos

-Siempre que borremos un radiador nos encargaremos de borrar todas sus asociaciones (grupo, localización, medidas...)

-No permitiremos tener localizaciones en las que no tenga definido previamente (edificio, planta y sala)

-Si borramos el nombre de una programación borraremos todos los datos referentes a esa programación (temperaturas objetivo, fecha, asignaciones a radiadores...)

-No permitimos borrar una característica de un radiador mientras tengamos algún radiador con esa característica asignada.



5. Conclusiones y líneas futuras

5.1 Conclusiones personales

Tras la finalización del proyecto se hizo una valoración de todo lo realizado, la opinión general fue buena ya que la adaptación a la empresa y al lenguaje fueron fáciles y rápidas. El producto desarrollado cumple las necesidades del cliente y el desarrollador(yo) ha aprendido un nuevo lenguaje de programación, el cual a día de hoy puede ser de gran utilidad ya que es un lenguaje orientado a la Web y este es uno de los sectores que mas está creciendo.

Otra aspecto a aprender es la diferencia de hacer un programa para clase o para una empresa, en los programas desarrollados en clase solo te preocupas de cubrir las necesidades del momento, pero si estas en una empresa tienes que tener una visión de cara al futuro en la que tendrás que realizar cambios. Esto supone hacer un proyecto flexible a los cambios e intentar anticiparte a futuras situaciones que nos puedan crear conflictos. No se puede crear “apaños” para solucionar un error puntual y crear otro error que en el momento no nos afecta.

Otra de las cosas interesantes del proyecto ha sido interactuar con otras personas para la realización del proyecto, ya que mi proyecto se unía aun proyecto previo que aun sigue en desarrollo. Al haber otras personas en el proyecto tienes que intentar trabajar lo más rápido posible para no ralentizar el trabajo del otro.

Importante dejar una documentación lo más clara posible y hacer las cosas de la forma más intuitiva posible ya que la otra persona no piensa como tú.

Al realizar el proyecto partiendo desde cero nos ha permitido conocer la dificultad de otros proyectos de mayor magnitud, ya que en nuestro proyecto que lo podemos considerar pequeño nos han surgido variedad de problemas y dificultades, en ese momento es cuando te das cuenta de la importancia de una buena planificación

A pesar de no haber estudiado nunca Silverlight nos a sido fácil la adaptación al nuevo lenguaje porque durante la carrera hemos tenido 2 asignaturas en las que el lenguaje estudiado estaba orientado a objetos(C# que es un lenguaje con el que



programa en silverlight también es orientado a objetos) y esto nos ha supuesto tener la mentalidad predispuesta a esta nueva forma de programar.

Por último agradecer a la empresa LATOR S.L y más en concreto a D.Patxi Tornaria por confiar en mí para la realización de este proyecto, confiando desde el primer día en mí y ofreciéndome una gran flexibilidad en el trabajo.

También dar las gracias a mi tutor del proyecto Jesus Villadangos ya que me ha ayudado mucho en el proyecto. Su ayuda no solo ha sido meramente en aspectos técnicos, también ha intervenido en las reuniones con la empresa y ha aportado otro punto de vista en el que trataba de minimizar la dificultad de las tareas propuestas por la empresa.



5.2 Conclusiones Técnicas

Hemos aprendido la dificultad de trabajar con un lenguaje Web como es Silverlight ya que tenemos una estructura cliente-servidor. Esto dificulta todas las cosas ya que el acceso a los datos no es tan sencillo, tenemos que crear una conexión ADO Entity con la que establecemos la unión entre el cliente y el servidor y luego tenemos que usar el Domain Service que será el encargado de mostrar los datos del servidor al cliente. Esto supuso un cambio en nuestra mentalidad de trabajo puesto que siempre que realizábamos alguna conexión la hacíamos de forma directa.

Ver que toda cosa por nueva o rara que pienses que estas haciendo, habrá alguien que ya la habrá desarrollado.

Adquisición de conocimientos de Silverlight, lo que implica conocimientos en xaml y C#, siendo este segundo muy importante en mi día a día, ya que toda la comunicación con el cliente (comunicador) la realizamos en C#.

Hemos aprendido a usar la herramienta Visual Studio, dicha herramienta es muy potente en la que podemos programar en distintos lenguajes, esta herramienta dispone de la tecnología inteligencia.



5.3 Lineas Futuras

Una vez que ha llegado la fecha de presentación del proyecto ya se han cumplido tanto las necesidades básicas del proyecto como las expectativas de la empresa, pero durante la realización del mismo nos han surgido una serie de ideas que podrían enriquecer mucho esta aplicación.

Algunas de ellas serian:

- Incluir gráficos en la pantalla de medidas, de esta forma seria mucho más fácil la detección de errores. El grafico constaría de dos comparativas la temperatura objetivo y las temperaturas reales reflejadas.

- A la hora de mostrar las temperaturas de la tablas de las programaciones definir unos colores que representasen diferentes rangos de temperaturas. Por ejemplo, azul para temperaturas de 15-17 , verde de 18-19 , naranja 20- 22 y rojo por encima de 22. De esta manera a simple vista veríamos cuando la calefacción va a estar trabajando a máxima potencia o va a estar apagada.

- Se pretende crear unos estándares en las temperaturas, temperatura confort, frío, económica...

- Otra cosa a implementar será crear usuarios con roles. Vamos a definir 3 niveles de acción, tendremos un usuario administrador general, mantenedor de la aplicación y bedel.

Administrador General: Este será el que puede tener acceso a todos los datos de todos los centros en los que esté implantado la aplicación. Será el encargado de definir todas las características de los radiadores.

Mantenedor: Será el encargado de dar de alta los radiadores, programaciones, grupos y localizaciones de ese centro. Esta persona tendrá que tener unos conocimientos básicos de informática y conocer bien la aplicación

Bedel: La tarea de está consistirá en poder asignar las programaciones creadas los radiadores que desee. En su mayoría todo estará creado y asignado por el



mantenedor pero siempre se pueden dar imprevistos y necesitamos una persona que puede solucionar sin tener que llamar al técnico

-En los inicios de la aplicación se tenía la intención de implantar la aplicación en un pantalla táctil, si esto se llevase a cabo tendríamos que modificar la aplicación para cambiar el tamaño tanto de los botones como el de las imágenes, ya que no es lo mismo trabajar con el ratón que trabajar con el dedo.

Durante bastante tiempo se estuvo trabajando con el DeepZoom, este programa nos ofrecía la capacidad de hacer zoom sobre una imagen de una manera muy atractiva, tuvimos que parar esta parte de la implementación porque la empresa prefería la funcionalidad a lo visible y se quedó un poco de lado. Sería interesante retomarlo en un futuro si finalmente se introduce en un soporte táctil, puesto que este zoom es mucho mas visual que el que hemos usado ahora.



6. Bibliografía

En este apartado se mencionan los libros, documentos, direcciones Web, etc. que se han tenido como referencia tanto para desarrollar la aplicación como para realizar la presente memoria.

6.1. Libros de texto

- Beginning Silverlight 4 in Csharp
- Silverlight 4 in Action
- Apress Pro Silverlight 3 in C 2009
- Programación en Silverlight 2.0, Cuaderno técnico nº 9



6.2 Direcciones web

Tratamiento de Fehcas

<http://www.canalvisualbasic.net/foro/visual-basic-net/convertir-datos-de-textbox-date-6581/>

http://foro.elhacker.net/net/comparar_horas_fechas_c_solucionado-t235042.0.html

<http://www.developerfusion.com/article/84338/making-usb-c-friendly/>

<http://www.canalvisualbasic.net/foro/visual-basic-6-0/valores-datetime-8754/>

Tratamiento de ficheros

http://www.elguille.info/NET/dotnet/leer_escribir_ficheros_texto.htm

Envio de datos USB

<http://www.dreamincode.net/forums/topic/111562-usb-port-handling/>

<http://www.dreamincode.net/forums/topic/182195-help-with-sending-data-using-usb-connection/>

<http://social.msdn.microsoft.com/forums/es-ES/vcses/thread/b95676f5-e07f-478b-b3b1-f0abb0d9e94b/>

Tratamiento de Excel

<http://geeks.ms/blogs/aruiz/archive/2007/01/25/accediendo-a-datos-de-excel-desde-c-con-oledb.aspx>

<http://social.msdn.microsoft.com/Forums/en-US/vsto/thread/b6e8a28c-6760-4e86-a1aa-e2ce9ec36380/>



<http://www.davidhayden.com/blog/dave/archive/2006/05/31/2976.aspx>

<http://www.forgottenprojects.com/excel-insertar-datos-c/>

Documentación de la memoria

http://es.wikipedia.org/wiki/Adobe_Flash

http://es.wikipedia.org/wiki/Rich_Internet_Applications

http://es.wikipedia.org/wiki/C_Sharp

<http://msdn.microsoft.com/es-es/library/cc903925%28v=VS.95%29.aspx>

<http://msdn.microsoft.com/es-es/silverlight/bb187401.aspx>

<http://msdn.microsoft.com/es-es/library/ee707344%28v=vs.91%29.aspx>

Aplicación Silverlight para sistema de gestión de la calefacción del Colegio de la Ultzama

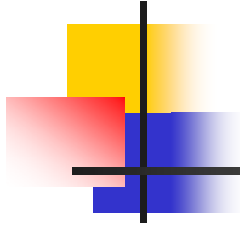
Jorge Sanz Díaz



Aplicación Silverlight para sistema de gestión de la calefacción del Colegio de la Ultzama

- **Introducción a la empresa**
- Antecedentes
- Requisitos del Sistema
- Análisis
- Diseño e Implementación
- Pruebas
- Conclusiones
- Líneas Futuras

Aplicación Silverlight para sistema de gestión de la calefacción del Colegio de la Ultzama



- Surge como proyecto innovador
- Principales sectores que abarca
 - Diseño Industrial
 - Naves Industriales
 - *Desarrollo Sostenible*
 - *Desarrollo de proyectos de I+D*
 - Fabricación



Aplicación Silverlight para sistema de gestión de la calefacción del Colegio de la Ultzama

- Introducción a la empresa
- **Antecedentes**
- Requisitos del Sistema
- Análisis
- Diseño e Implementación
- Pruebas
- Conclusiones
- Líneas Futuras



Aplicación Silverlight para sistema de gestión de la calefacción del Colegio de la Ultzama

- Surge la necesidad de individualizar los radiadores
- Creación válvula
- Control sobre el consumo
- Distribución en el colegio
- Protocolo
- 2 Etapas de desarrollo

Aplicación Silverlight para sistema de gestión de la calefacción del Colegio de la Ultzama

- 1ª Etapa
 - Interfaz Gráfico Amigable
 - Diagrama Sectores
 - Táctil
 - Deep Zoom
- 2ª Etapa
 - Interés de LATOR SL
 - Cambio radical (estético por funcional)
 - 2 Módulos
 - Gestión BD
 - Comunicación USB



Aplicación Silverlight para sistema de gestión de la calefacción del Colegio de la Ultzama

- Introducción a la empresa
- Antecedentes
- **Requisitos del Sistema**
- Análisis
- Diseño e Implementación
- Pruebas
- Conclusiones
- Líneas Futuras



Aplicación Silverlight para sistema de gestión de la calefacción del Colegio de la Ultzama

- Aplicación Web
- Gestión de Programaciones
- Gestión de Parámetros Entrada/Salida

Aplicación Silverlight para sistema de gestión de la calefacción del Colegio de la Ultzama

- Aplicación Web
 - Elección del Lenguaje
 - Silverlight 4 (plug-in)
 - Rich Internet Applications
 - Aplicaciones de escritorio
 - Navegador Web
 - Web recarga continua de datos
 - Cliente Servidor
 - .NET
 - Visual Studio
 - Expression Estudio
 - Code-Behind (C#)
 - SQL Server 2008



Aplicación Silverlight para sistema de gestión de la calefacción del Colegio de la Ultzama

- Aplicación Web
 - WCF Ria Services
 - Silverlight es una tecnología cliente-servidor.
 - n niveles en aplicaciones RIA.
 - Combina Entity Framework (ORM) con WCF (Windows Communication Foundation)



Aplicación Silverlight para sistema de gestión de la calefacción del Colegio de la Ultzama

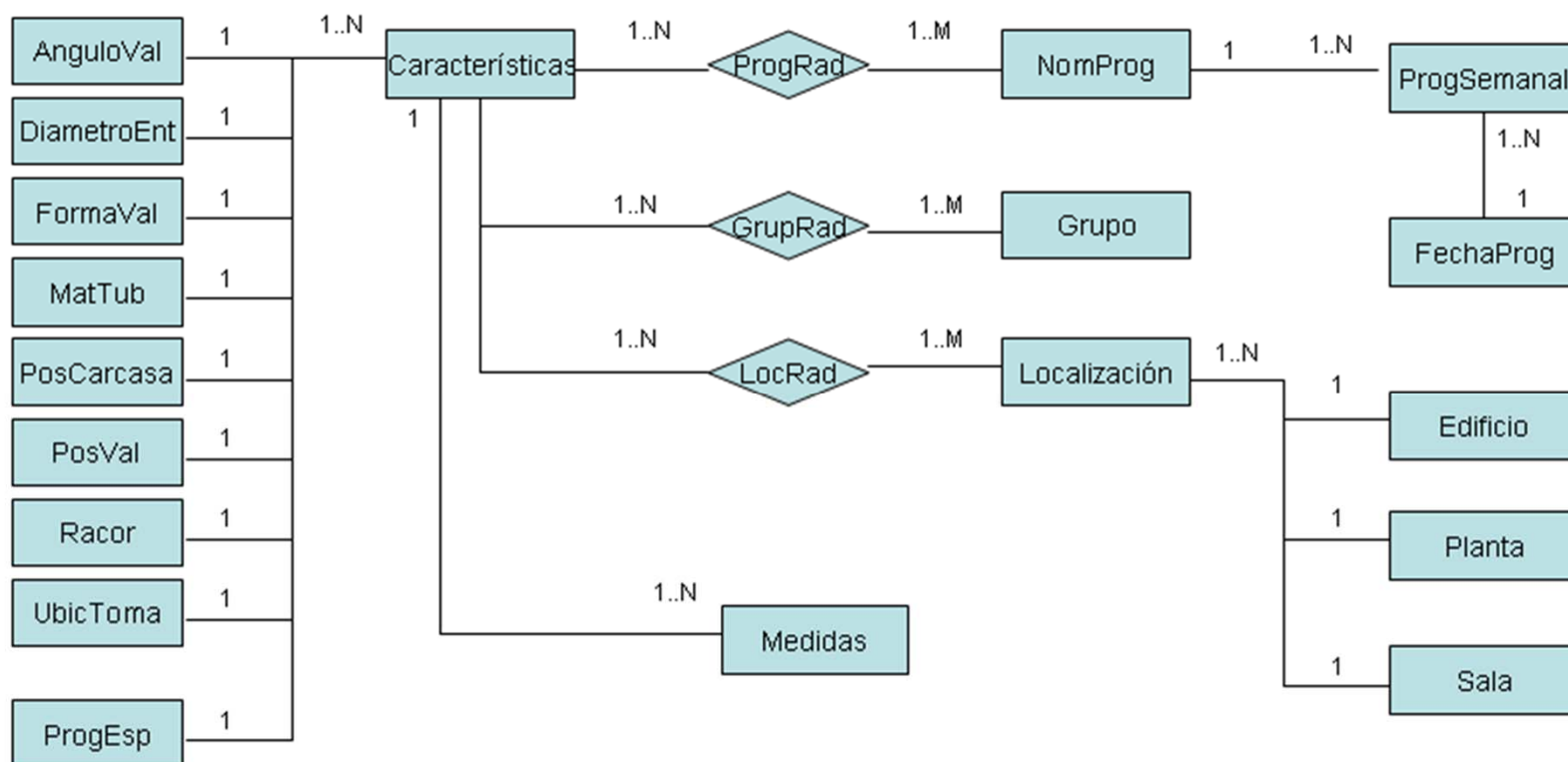
- Introducción a la empresa
- Antecedentes
- Requisitos del Sistema
- **Análisis**
- Diseño e Implementación
- Pruebas
- Conclusiones
- Líneas Futuras



Aplicación Silverlight para sistema de gestión de la calefacción del Colegio de la Ultzama

- **Modelo E-R**
- **Protocolo de Comunicación WinUSB**

Aplicación Silverlight para sistema de gestión de la calefacción del Colegio de la Ultzama

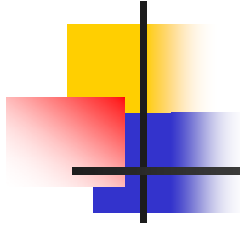




Aplicación Silverlight para sistema de gestión de la calefacción del Colegio de la Ultzama

- Modelo E-R
- **Protocolo de Comunicación WinUSB**

Aplicación Silverlight para sistema de gestión de la calefacción del Colegio de la Ultzama



- Driver WinUSB
- Comunicación host-device
- Problema inicio comunicación
- Tipos transferencia
 - Transferencia de Control
 - Transferencia Bulk
 - Transferencia Interrupción
 - Transferencia Asíncrona

Aplicación Silverlight para sistema de gestión de la calefacción del Colegio de la Ultzama

- 5 tipos de paquetes
 - 0x10
 - 0x20
 - 0x30
 - 0x40
 - 0x50
- Protocolo de comunicación



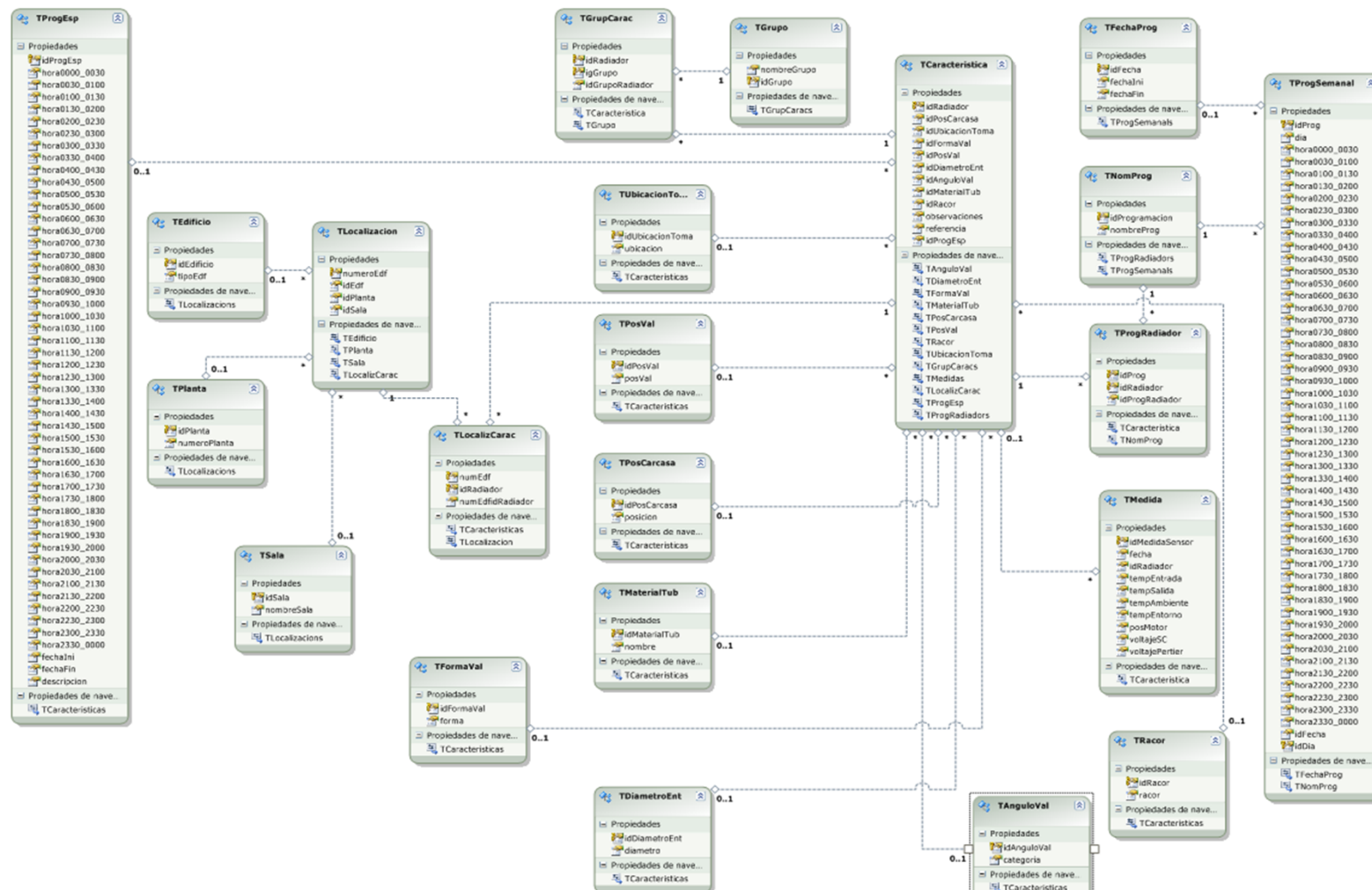
Aplicación Silverlight para sistema de gestión de la calefacción del Colegio de la Ultzama

- Introducción a la empresa
- Antecedentes
- Requisitos del Sistema
- Análisis
- **Diseño e Implementación**
- Pruebas
- Conclusiones
- Líneas Futuras



Aplicación Silverlight para sistema de gestión de la calefacción del Colegio de la Ultzama

- **Diseño E-R**
- Diseño e implementación de Casos de Uso

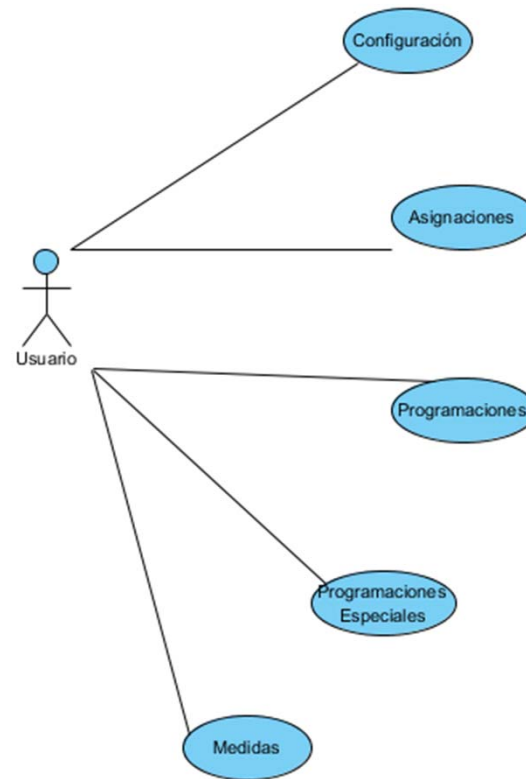




Aplicación Silverlight para sistema de gestión de la calefacción del Colegio de la Ultzama

- Diseño E-R
- **Diseño e implementación de Casos de Uso**

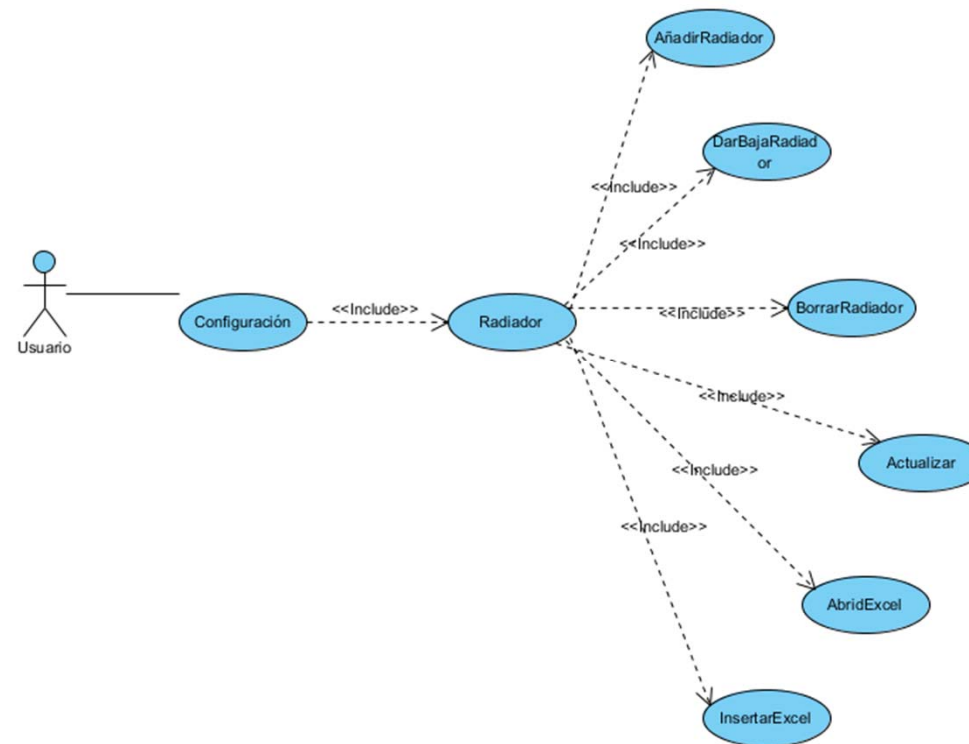
Aplicación Silverlight para sistema de gestión de la calefacción del Colegio de la Ultzama



Aplicación Silverlight para sistema de gestión de la calefacción del Colegio de la Ultzama



Aplicación Silverlight para sistema de gestión de la calefacción del Colegio de la Ultzama

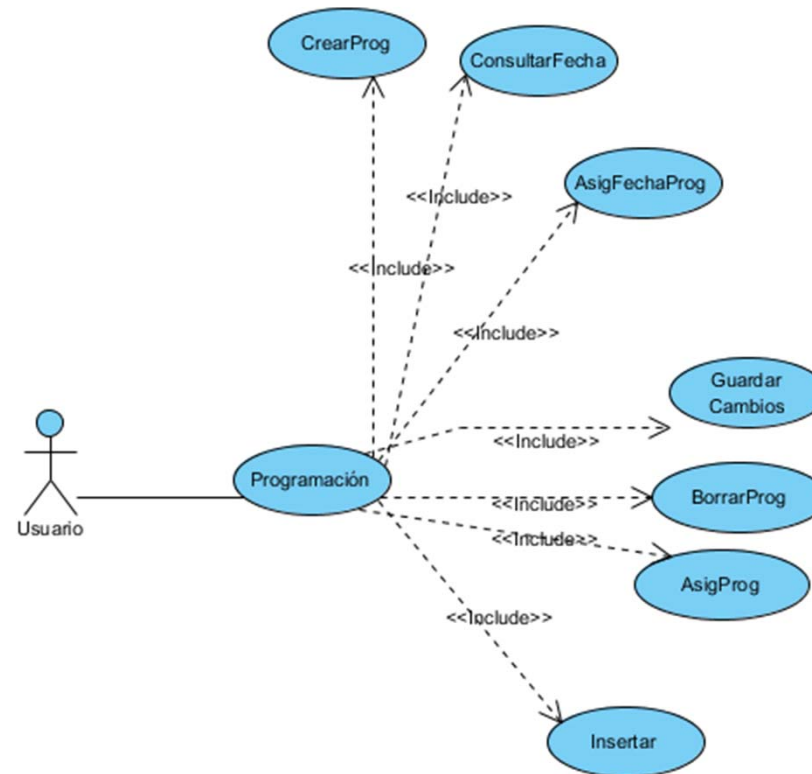


Aplicación Silverlight para sistema de gestión de la calefacción del Colegio de la Ultzama

Colegio Ultzama Home **Configuración** Asignaciones Programaciones Programaciones Especiales Medidas login

Radiador	Características Radiador		Localización	Partes Localización		Grupos						
Radiador	Angulo Val	Diametro Ent	Forma Val	Material Tub	Pos Carcasa	Pos Val	Racor	Ubicacion Toma	Observaciones	Referencia	Numero Serie	
2	180	34	E	PM	B	I	34	P	ninguna	fasda	1	AñadirRadiador
3	180	34	E	PM	A	D	34	P	ninguna	fasda	2	
15	270	34	E	CM	A	D	12	P	esto es una buer	prueba	3	DarBajaRadiador
16	270	34	E	PM	A	I	12	P	prueba 1 seg	bueno	4	
39	180	34	E	CM	A	I	12	P	Primer radiador r	NDFSWEFS	5	BorrarRadiador
40	270	34	E	CM	B	D	34	P	asas	SSADASD	6	
43	180	34	E	CM	A	D	34	P	Hola k tal	KJMNJI	7	Actualizar
44	180	34	E	CM	A	D	12	P		ESSFCE	8	
46	270	34	E	CM	A	D	12	P			9	
3	254	0.5	E	H	A	D	0.5	S		ASED-0.5-254H-0.5	123456789012	
4	254	0.5	E	H	A	D	0.5	S		ASED-0.5-254H-0.5	123456789013	AbrirExcel
												InsertarExcel

Aplicación Silverlight para sistema de gestión de la calefacción del Colegio de la Ultzama



Aplicación Silverlight para sistema de gestión de la calefacción del Colegio de la Ultzama

Colegio Ultzama Home Configuración Asignaciones **Programaciones** Programaciones Especiales Medidas login

Programacion: CrearProg

Fecha: ConsultarFechas AsigFechaProg Nombre de la Programación: ProgInvierno FechaInicio: 02/06/2011 FechaFin: 30/06/2011

	Id Fecha	Id Prog	00:00 ->	00:30 ->	01:00 ->	01:30 ->	02:00 ->	02:30 ->	03:00 ->	03:30 ->	04:00 ->	04:30 ->	05:00 ->	05:30 ->	06:00 ->	06:30 ->	07
Lunes	1	65	12	12	25	12	12	12	15	15	15						
Martes	1	65	15	15	15	15	15	15	15	15	15						
Miércoles	1	65	15	15	15	15	15	15	15	15	15						
Jueves	1	65	15	15	15	15	8	9	10	15	15						
Viernes	1	65	15	15	12	11	3	2	4	5	15						
Sábado	1	65	15	15	15	15	15	15	15	15	15						
Domingo	1	65	15	15	15	15	15	15	15	15	15						

Guardar Cambios Borrar Programación Asignar Programación

Lunes De: a °

Martes De: a °

Miércoles De: a °

Jueves De: a °

Viernes De: a °

Sábado De: a °

CrearFechaWin

Id Fecha	Fecha Inicio	Fecha Fin
1	02/06/2011	30/06/2011
5	21/06/2011	23/06/2011
13	02/08/2011	23/08/2011

CrearFecha Modificar BorrarFecha Aceptar Cancelar



Aplicación Silverlight para sistema de gestión de la calefacción del Colegio de la Ultzama

- Introducción a la empresa
- Antecedentes
- Requisitos del Sistema
- Análisis
- Diseño e Implementación
- **Pruebas**
- Conclusiones
- Líneas Futuras

Aplicación Silverlight para sistema de gestión de la calefacción del Colegio de la Ultzama

- Comunicación entre módulos
- Comunicación con el device
 - Correcto cumplimiento del protocolo creado
 - 4 Pasos
 - Inserciones
 - Consultas programación enviar
- Pruebas de unicidad
 - No solapamiento programaciones
 - Borrar Radiador, eliminar asociaciones
 - Características radiador



Aplicación Silverlight para sistema de gestión de la calefacción del Colegio de la Ultzama

- Introducción a la empresa
- Antecedentes
- Requisitos del Sistema
- Análisis
- Diseño e Implementación
- Pruebas
- **Conclusiones**
- Líneas Futuras

Aplicación Silverlight para sistema de gestión de la calefacción del Colegio de la Ultzama

- Diferencia programas de clase y empresa
- Interactuar con otras personas
- Dificultad crear proyecto desde 0
- Dificultad usar un lenguaje Web
- Importante saber moverse en la red, muchos problemas ya están resueltos



Aplicación Silverlight para sistema de gestión de la calefacción del Colegio de la Ultzama

- Introducción a la empresa
- Antecedentes
- Requisitos del Sistema
- Análisis
- Diseño e Implementación
- Pruebas
- Conclusiones
- **Líneas Futuras**



Aplicación Silverlight para sistema de gestión de la calefacción del Colegio de la Ultzama

- Incluir gráficos
- Asignar colores a las temperaturas
- Usuarios: 3 niveles diferentes
 - Administrador
 - Mantenedor
 - Bedel
- Atributos posición X e Y para los radiadores
- Volver al enfoque de la pantalla táctil